
django-qartez Documentation

Release 0.5

Artur Barseghyan <artur.barseghyan@gmail.com>

Sep 27, 2017

Contents

| | | |
|----------|--|-----------|
| 1 | Installation | 3 |
| 1.1 | 1. Install | 3 |
| 1.2 | 2. Add <i>qartez</i> to your <code>INSTALLED_APPS</code> | 3 |
| 2 | Usage and examples | 5 |
| 2.1 | foo/sitemap.py | 5 |
| 2.2 | urls.py | 6 |
| 2.3 | foo/models.py | 7 |
| 2.4 | foo/views.py | 7 |
| 2.5 | foo/urls.py | 8 |
| 3 | License | 9 |
| 4 | Support | 11 |
| 5 | Author | 13 |
| 6 | Documentation | 15 |
| 6.1 | qartez Package | 15 |
| 6.2 | conf Module | 17 |
| 6.3 | settings Module | 17 |
| 6.4 | views Module | 17 |
| 7 | Indices and tables | 19 |
| | Python Module Index | 21 |

This app aims to provide the missing XML sitemaps for Django. At the moment the following XML sitemaps are implemented:

- `qartez.ImagesSitemap`: XML images sitemaps according to the specs <http://www.google.com/support/webmasters/bin/answer.py?answer=178636>
- `qartez.StaticSitemap`: Sitemap for static pages. Add named patterns or URLs to the sitemap to have it nicely displayed in a separate static XML sitemap.
- `qartez.RelAlternateHreflangSitemap`: Sitemaps: `rel="alternate" hreflang="x"` implementation. Read the specs the specs here <http://support.google.com/webmasters/bin/answer.py?hl=en&answer=2620865>

1. Install

Latest stable version on pypi:

```
$ pip install django-qarte
```

Latest stable version from source:

```
$ pip install -e hg+http://bitbucket.org/barseghyanartur/django-qarte@stable#egg=django-qarte
```

2. Add *qarte* to your `INSTALLED_APPS`

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


CHAPTER 2

Usage and examples

We have an imaginary foo app.

The full source code of the example below is at <http://bitbucket.org/barseghyanartur/django-qartez/src> (see the *example* directory).

foo/sitemap.py

```
>>> from django.contrib.sitemaps import Sitemap
>>>
>>> from qartez 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 -----
>>> # 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()
```

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

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, \
>>>                                         default=datetime.datetime.now())
>>>
>>>     # 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 __unicode__(self):
>>>         return self.title
>>>
>>>     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 ''
```

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))
```

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 3

License

GPL 2.0/LGPL 2.1

CHAPTER 4

Support

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

CHAPTER 5

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

qartez Package

class `qartez.__init__.ImagesSitemap` (*info_dict*, *priority=None*, *changefreq=None*)

Bases: `django.contrib.sitemaps.GenericSitemap`

Class for image sitemap. Implemented according to specs specified by Google <http://www.google.com/support/webmasters/bin/answer.py?answer=178636>

Example

```
>>> from qartez import ImagesSitemap
>>>
>>> foo_item_images_info_dict = {
>>>     'queryset': FooItem._default_manager.exclude(image=None), # queryset
>>>     'image_location_field': 'image', # image location
>>>     'image_title_field': 'title', # image title
>>>     'location_field': 'get_absolute_url' # an absolute URL of the page
>>>                                     # where image is shown
>>> }
>>>
>>> foo_item_images_sitemap = {
>>>     'foo_item_images': ImagesSitemap(foo_item_images_info_dict,    >>>
↵                                     priority=0.6),
>>> }
```

image_caption (*item*)

Gets image caption.

image_geo_location (*item*)

Gets image geo location.

image_license (*item*)

Gets image geo location.

image_location (*item*)

Gets image location.

image_title (*item*)

Gets image title.

location (*item*)

Gets image location URL.

class qartez.__init__.StaticSitemap(*args, **kwargs)

Bases: django.contrib.sitemaps.Sitemap

Sitemap for static pages. See constructor docstring for list of accepted (additional) arguments.

Example

```
>>> from qartez import StaticSitemap
>>> service_sitemap = StaticSitemap(priority=0.1, changefreq='never')
>>> service_sitemap.add_named_pattern('blog.welcome')
>>> service_sitemap.add_named_pattern('feedback.contact')
>>>
>>> content_types_sitemap = StaticSitemap(priority=1.0, changefreq='daily')
>>> content_types_sitemap.add_named_pattern('blog.browse') # Homepage
>>> content_types_sitemap.add_named_pattern(
>>>     'blog.browse', kwargs={'content_type': 'articles'}
>>> ) # Articles
>>> content_types_sitemap.add_named_pattern(
>>>     'blog.browse', kwargs={'content_type': 'downloads'}
>>> ) # Downloads
```

add_named_pattern (viewname, urlconf=None, args=[], kwargs=None, lastmod=None, changefreq=None, priority=None)

Ads a named pattern to the items list.

Parameters

- **viewname** (*str*) –
- **urlconf** –
- **args** (*list*) –
- **kwargs** (*dict*) –
- **lastmod** –
- **changefreq** (*str*) –
- **priority** (*float*) –

add_url (url, lastmod=None, changefreq=None, priority=None)

Adds a URL to the items list.

Parameters

- **url** (*str*) –
- **lastmod** –
- **changefreq** (*str*) –
- **priority** (*float*) –

get_urls (*args, **kwargs)

Make sure nothing breaks if some URL is unresolvable.

Return list**items()**

Returns sitemap items.

Return list**class** qartez.__init__.RelAlternateHreflangSitemap

Bases: django.contrib.sitemaps.Sitemap

Sitemaps: rel="alternate" hreflang="x" implementation.

Read the specs the specs here <http://support.google.com/webmasters/bin/answer.py?hl=en&answer=2620865>IMPORTANT: When you use this class you have to override the `alternate_hreflangs` method in your sitemap class.**Example**

```
>>> from qartez import RelAlternateHreflangSitemap
>>>
>>> class ArticleSitemap(RelAlternateHreflangSitemap):
>>>     def alternate_hreflangs(self, obj):
>>>         return [('en-us', obj.alternative_object_url),]
```

alternate_hreflangs (*item*)

You should override the “alternate_hreflangs” method in your sitemap class.

conf Module

qartez.conf.get_setting (*setting*, *override=None*)

Get a setting from qartez conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

settings Module

Override the following values in your global settings module by adding *QARTEZ_* prefix to the values.When it comes to importing the values, import them from `qartez.settings` module (without *QARTEZ_* prefix).**PREPEND_LOC_URL_WITH_SITE_URL**: When set to True, current site’s domain is prepended to the location URL.**PREPEND_IMAGE_LOC_URL_WITH_SITE_URL**: When set to True, current site’s domain is prepended to the image location URL.**CHANGEFREQ**: Valid changefreq values according to the specs <http://www.sitemaps.org/protocol.html>

views Module

qartez.views.render_images_sitemap (*request*, *sitemaps*, *section=None*, *template_name='qartez/images_sitemap.xml'*)

Renders images sitemap.

Parameters

- **request** (*django.http.HttpRequest*) –

- `sitemaps` –
- `seccion` –
- `template_name` (*str*) –

Return `django.http.HttpResponse`

CHAPTER 7

Indices and tables

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

q

`qartez.__init__`, [15](#)

`qartez.conf`, [17](#)

`qartez.settings`, [17](#)

`qartez.views`, [17](#)

A

`add_named_pattern()` (qartez.__init__.StaticSitemap `StaticSitemap` (class in qartez.__init__), 16 method), 16
`add_url()` (qartez.__init__.StaticSitemap method), 16
`alternate_hreflangs()` (qartez.__init__.RelAlternateHreflangSitemap method), 17

G

`get_setting()` (in module qartez.conf), 17
`get_urls()` (qartez.__init__.StaticSitemap method), 16

I

`image_caption()` (qartez.__init__.ImagesSitemap method), 15
`image_geo_location()` (qartez.__init__.ImagesSitemap method), 15
`image_license()` (qartez.__init__.ImagesSitemap method), 15
`image_location()` (qartez.__init__.ImagesSitemap method), 15
`image_title()` (qartez.__init__.ImagesSitemap method), 16
`ImagesSitemap` (class in qartez.__init__), 15
`items()` (qartez.__init__.StaticSitemap method), 17

L

`location()` (qartez.__init__.ImagesSitemap method), 16

Q

`qartez.__init__` (module), 15
`qartez.conf` (module), 17
`qartez.settings` (module), 17
`qartez.views` (module), 17

R

`RelAlternateHreflangSitemap` (class in qartez.__init__), 17
`render_images_sitemap()` (in module qartez.views), 17