

---

# **django-siteblocks Documentation**

*Release 0.5.0*

**Igor 'idle sign' Starikov**

April 17, 2015



<b>1</b>	<b>Requirements</b>	<b>3</b>
<b>2</b>	<b>Table of Contents</b>	<b>5</b>
2.1	Getting started . . . . .	5
2.2	Dynamic siteblocks . . . . .	6
2.3	Thirdparty applications support . . . . .	7
<b>3</b>	<b>Get involved into django-siteblocks</b>	<b>9</b>
<b>4</b>	<b>Also</b>	<b>11</b>



<http://github.com/idlesign/django-siteblocks>

*django-siteblocks is a reusable application for Django to build blocks of static or dynamic data that could be used in templates.*

It allows you to describe data that doesn't clearly belong to any application in your project in terms of static or dynamic blocks, that could be rendered in certain places on site pages. These blocks are addressed in templates by their aliases.



---

## Requirements

---

1. Python 2.7+ or 3.3+
2. Django 1.4+
3. Django Admin contrib enabled (optional)
4. South 1.0+ (for automatic DB migrations; not required for Django 1.7+)





---

## Table of Contents

---

### 2.1 Getting started

- Add the **siteblocks** application to `INSTALLED_APPS` in your settings file (usually 'settings.py').
- Use `> python manage.py syncdb` command to install app tables in DB (`> python manage.py migrate` for Django 1.7+)
- Make sure `TEMPLATE_CONTEXT_PROCESSORS` in your settings file has `django.core.context_processors.request` if you want to use static blocks created in Django Admin. For Django 1.8+: `django.template.context_processors.request` should be defined in `TEMPLATES/OPTIONS/context_processors`.

**Warning:** Those, who are using South <1.0 for migrations with Django <1.7, add this into settings file:

```
SOUTH_MIGRATION_MODULES = {
    'siteblocks': 'siteblocks.south_migrations',
}
```

#### 2.1.1 Quick example

**Note:** This example covers only static siteblocks. More advanced technique can be found in *Dynamic siteblocks* section.

Let's say we need random quotes block.

1. Add `{% siteblock "my_quotes" %}` tag where you need it in templates. Here `my_quotes` is the alias of a block.

```
{% extends "_base.html" %}
{% load siteblocks %}

{% block sidebar %}
    <div class="quote">
        {% siteblock "my_quotes" %}
    </div>
{% endblock %}
```

**Note:** You can use `{% siteblock "my_quotes" as my_var %}` tag variation to put block contents into a template context variable.

2. Go to Django Admin site and add several siteblocks with quotes aliased *my\_quotes*.

*Note that you can render different sets of quotes on different pages (URLs or views).*

Random quotes are here. You're done.

## 2.2 Dynamic siteblocks

*Dynamic* siteblocks instead of *Static* ones allows complex logic to build siteblock contents.

Siteblocks content in this case is not defined using Django Admin contrib, but generated by an ordinary Python function registered as a siteblock.

Use `register_dynamic_block()` function to register a callable that produces contents for a dynamic block.

Such a callable will get the following kwargs when called:

- `block_alias` - block alias,
- `block_context` - template context for block

Let's implement quotes siteblock similar to that from *Getting started*, but *dynamic* this time.

1. Add `{% siteblock "my_quotes" %}` tag where you need it in templates. Here `my_quotes` is the alias of a block.

```
{% extends "_base.html" %}
{% load siteblocks %}

{% block sidebar %}
    <div class="quote">
        {% siteblock "my_quotes" %}
    </div>
{% endblock %}
```

2. Define and register a dynamic siteblock.

*# Put the following code somewhere where it'd be triggered as expected. E.g. in app view.py.*

```
from random import choice
# Import the register function.
from siteblocks.siteblocksapp import register_dynamic_block
```

*# The following function will be used as a block contents producer.*

```
def get_quote(**kwargs):
    quotes = [ # From Terry Pratchett's Discworld novels.
        'Ripples of paradox spread out across the sea of causality.',
        'Early to rise, early to bed, makes a man healthy, wealthy and dead.',
        'Granny had nothing against fortune-telling provided it was done badly by people wit',
        'Take it from me, there\'s nothing more terrible than someone out to do the world a',
        'The duke had a mind that ticked like a clock and, like a clock, it regularly went c',
        'Most gods find it hard to walk and think at the same time.',
        'They didn\'t have to be funny - they were father jokes',
        'Speak softly and employ a huge man with a crowbar.',
    ]
    return choice(quotes)
```

```
# And we register our siteblock.
register_dynamic_block('my_quotes', get_quote)
```

## 2.3 Thirdparty applications support

Here belongs some notes on thirdparty Django applications support in *siteblocks*.

### 2.3.1 django-modeltranslation

<https://pypi.python.org/pypi/django-modeltranslation/>

You can use *modeltranslation* to make your site blocks language dependent.

This will require setting *SITEBLOCKS\_I18N\_SUPPORT* to **True** in *settings.py* of your project.



---

## Get involved into django-siteblocks

---

**Submit issues.** If you spotted something weird in application behavior or want to propose a feature you can do that at <https://github.com/idlesign/django-siteblocks/issues>

**Write code.** If you are eager to participate in application development, fork it at <https://github.com/idlesign/django-siteblocks>, write your code, whether it should be a bugfix or a feature implementation, and make a pull request right from the forked project page.

**Translate.** If want to translate the application into your native language use Transifex: <https://www.transifex.com/projects/p/django-siteblocks/>.

**Spread the word.** If you have some tips and tricks or any other words in mind that you think might be of interest for the others — publish them.



---

**Also**

---

If the application is not what you want for user registration, you might be interested in considering the other choices — <https://www.djangopackages.com/grids/g/layout/>