

---

# **django-bootstrap-breadcrumbs** **Documentation**

*Release 0.6.3*

**Łukasz Mierzwa**

**Nov 12, 2017**



<b>1</b>	<b>Links</b>	<b>1</b>
<b>2</b>	<b>Requirements</b>	<b>3</b>
<b>3</b>	<b>Installation</b>	<b>5</b>
<b>4</b>	<b>Declaring breadcrumbs</b>	<b>7</b>
4.1	{% breadcrumb %} .....	7
4.2	{% breadcrumb_safe %} .....	8
4.3	{% breadcrumb_raw %} .....	8
4.4	{% breadcrumb_raw_safe %} .....	8
4.5	{% breadcrumb_for %} .....	8
<b>5</b>	<b>Rendering breadcrumbs</b>	<b>11</b>
<b>6</b>	<b>Full examples</b>	<b>13</b>
<b>7</b>	<b>Changelog</b>	<b>15</b>
<b>8</b>	<b>Contributors</b>	<b>17</b>



# CHAPTER 1

---

Links

---

Issue tracker: <https://github.com/primitive/bootstrap-breadcrumbs/issues>



## CHAPTER 2

---

### Requirements

---

- Python  $\geq 2.6$  ( $\geq 3.0$  supported since 0.6.1, requires Django  $\geq 1.5$ )
- Django  $\geq 1.4$
- Bootstrap 2.3, 3 or 4





## CHAPTER 3

---

### Installation

---

Just install it using pip (recommended):

```
pip install django-bootstrap-breadcrumbs
```

Or clone it from github:

```
git clone https://github.com/primitive/bootstrap-breadcrumbs.git
cd bootstrap-breadcrumbs
./setup.py install
```

After that make necessary changes to Django settings:

```
* add "django_bootstrap_breadcrumbs" to INSTALLED_APPS.
* make sure that TEMPLATE_CONTEXT_PROCESSORS contains "django.core.context_processors.
↳request" (for Django < 1.10)
* make sure that TEMPLATES->OPTIONS->context_processors contains "django.template.
↳context_processors.request" (for Django >= 1.10)
```



---

## Declaring breadcrumbs

---

There are currently three tags for adding breadcrumbs for pages (remember to use the tags inside a `{% block %}`):

### 4.1 `{% breadcrumb %}`

Syntax:

```
{% breadcrumb $label $viewname [*args] [**kwargs] %}
```

`label` - Breadcrumb link text.

`viewname` - Any string that can be resolved into a view url with django `reverse()` function or a django Model instance with implemented `get_absolute_url()` method.

`args` - Optional arguments to django's `reverse()` function.

`kwargs` - Optional keyword arguments to django's `reverse()` function.

`viewname` will be resolved into url using django `reverse()` function using:

```
url = resolve(viewname, args=args, kwargs=kwargs)
```

If the `viewname` cannot be resolved using `reverse()` then it will be rendered as is, so that static url's can be used in `{% breadcrumb %}` template tags.

Note that the label is escaped by default, so all HTML tags will be replaced. This is a protection for cases where the label can contain user provided content, for example a username. If the user would somehow put links or javascript into the label, he could expose any viewer to malicious code.

Example:

```
{% block breadcrumbs %}
    {% breadcrumb "Home" "/" %}
    {% breadcrumb "Users and groups" "users_and_groups_index" %}
{% endblock %}
```

label can be anything that django template will resolve:

```
{% block breadcrumbs %}
    {% breadcrumb user.username "user_profile" user.id %}
{% endblock %}
```

## 4.2 {% breadcrumb\_safe %}

It works just like `{% breadcrumb %}` but it doesn't do any escaping, use it if trusted HTML is required in the label.

Example:

```
{% block breadcrumbs %}
    {% breadcrumb_safe "<i class='icon-home'></i>Home" "/" %}
{% endblock %}
```

## 4.3 {% breadcrumb\_raw %}

By default breadcrumbs labels are translated using `gettext`, if the translation should be skipped `{% breadcrumb_raw %}` can be used (the label is still escaped). Available since 0.7.0 release.

## 4.4 {% breadcrumb\_raw\_safe %}

If the label should neither be escaped nor translated `{% breadcrumb_raw_safe %}` can be used. Available since 0.7.0 release.

## 4.5 {% breadcrumb\_for %}

Starting with 0.4.0 there is also block tag, usage:

```
{% breadcrumb_for $viewname [*args] [**kwargs] %}
    $label
{% endbreadcrumb_for %}
```

Any code can be used there, it won't be escaped in any way. It gives the possibility to fully control the label content and for example escape only parts of it.

---

**Note:** Since 0.7.0 final label part from `{% breadcrumb_for %}` is no longer translated, add `{% trans %}` tag if needed.

---

Examples:

```
{% breadcrumb_for site_index %}
    <i class='icon-home'></i>
    {% trans "Home" %}
{% endbreadcrumb_for %}
```

```
{% breadcrumb_for user_profile user.username %}
  <i class='icon-user'></i>
  {{ user.username }}
{% endbreadcrumb_for %}
```



---

## Rendering breadcrumbs

---

To render breadcrumbs as HTML use `{% render_breadcrumbs %}`.

---

**Important:** Remember that `{% render_breadcrumbs %}` tag must appear in template after all other breadcrumb tags.

---

Example:

```
{% block content %}
    {% render_breadcrumbs %}
{% endblock %}
```

Starting with 0.5.0 it's possible to use a custom template to integrate breadcrumbs with frameworks other than Bootstrap.

Example:

```
{% block content %}
    {% render_breadcrumbs "path/to/my/template.html" %}
{% endblock %}
```

Default template uses Bootstrap classes:

```
<ul class="breadcrumb">
  {% for url, label in breadcrumbs %}
    <li>
      {% ifnotequal forloop.counter breadcrumbs_total %}
        <a href="{{ url }}">{{ label|safe }}</a>
      {% else %}
        {{ label|safe }}
      {% endifnotequal %}
      {% if not forloop.last %}
        <span class="divider"></span>
      {% endif %}
    </li>
```

```
{% endfor %}
</ul>
```

- `breadcrumbs` - list of breadcrumbs elements, each element contains url and label
- `breadcrumbs_total` - total number of breadcrumbs elements

To use Bootstrap V3 template instead of V2, use:

```
{% block content %}
    {% render_breadcrumbs "django_bootstrap_breadcrumbs/bootstrap3.html" %}
{% endblock %}
```

Starting with 0.7.3 there's also V4 template:

```
{% block content %}
    {% render_breadcrumbs "django_bootstrap_breadcrumbs/bootstrap4.html" %}
{% endblock %}
```

Starting with 0.7.1 it's possible to set default template path in `settings.py` using `BREADCRUMBS_TEMPLATE='/my/template.html'`:

```
BREADCRUMBS_TEMPLATE = "django_bootstrap_breadcrumbs/bootstrap4.html"
```

Passing template path to `{% render_breadcrumbs %}` takes precedence over `BREADCRUMBS_TEMPLATE`.

With 0.6.0 a new template tag was added for clearing breadcrumbs list:

```
{% clear_breadcrumbs %}
```

It can be used if we want to replace current breadcrumbs list with new. It's mostly useful for adding breadcrumbs to error pages, such pages are rendered after parsing all view templates, so without clearing current list we would have doubled breadcrumbs. It's recommended to add `{% clear_breadcrumbs %}` to all root breadcrumbs (home links).



## CHAPTER 6

---

### Full examples

---

#### base.html:

```
{% load django_bootstrap_breadcrumbs %}

{% block breadcrumbs %}
    {% clear_breadcrumbs %}
    {% breadcrumb "Home" "/" %}
    {% breadcrumb "Users and groups" "users_and_groups_index" %}
{% endblock %}

{% block content %}
    {% render_breadcrumbs %}
{% endblock %}
```

#### users.html:

```
{% extends "base.html" %}

{% load django_bootstrap_breadcrumbs %}

{% block breadcrumbs %}
    {{ block.super }}
    {% breadcrumb "Users" "users.views.index" %}
{% endblock %}
```

#### profile.html:

```
{% extends "users.html" %}

{% load django_bootstrap_breadcrumbs %}

{% block breadcrumbs %}
    {{ block.super }}
    {% breadcrumb user "users.views.profile" user.username %}
{% endblock %}
```

Result:

```
Home / Users and groups / Users / John Doe
```

It's also possible to use properties.

profile.html:

```
{% extends "users.html" %}

{% load django_bootstrap_breadcrumbs %}

{% block breadcrumbs %}
    {{ block.super }}
    {% breadcrumb user.email "users.views.profile" user.username %}
{% endblock %}
```

500.html:

```
{% extends "users.html" %}

{% load django_bootstrap_breadcrumbs %}

{% block breadcrumbs %}
    {{ block.super }}
    {% breadcrumb "Internal error" "" %}
{% endblock %}
```

Result:

If everything **is** working:

```
Home / Users and groups / Users / john.doe@example.org
```

In case of internal error:

```
Home / Internal error
```

# CHAPTER 7

---

## Changelog

---

- 0.8.2 - no code changes since 0.8.1
- 0.8.1 - `{% render_breadcrumbs %}` will now pass context as dict for Django  $\geq$  1.8 since `RequestContext()` was deprecated (<https://docs.djangoproject.com/en/1.10/ref/templates/upgrading/>)
- 0.8 - lots of cleanups and improvements by Alexandre Macabies (zopieux)
- 0.7.3 - added bootstrap v4 template (JP-Ellis)
- 0.7.2 - fixed context passing in `render_breadcrumbs()` (JeLoueMonCampingCar)
- 0.7.1 - added support for setting default template path in `settings.py` using `BREADCRUMBS_TEMPLATE='my/template.html'` (gdebure)
- 0.7.0 - added `breadcrumb_raw` and `breadcrumb_raw_safe`, label in `breadcrumb_for` is no longer translated
- 0.6.3 - added support for passing kwargs to breadcrumb tags
- 0.6.2 - license changed to MIT
- 0.6.1 - python3 support
- 0.6.0 - added `clear_breadcrumbs` template tag
- 0.5.5 - handle resolver errors so that breadcrumbs might be used in 404 or 500 template
- 0.5.4 - warn if request object is missing from context but don't raise error
- 0.5.3 - support for namespaced urls (edavis)
- 0.5.2 - added bootstrap v3 template
- 0.5.1 - added missing template to the package
- 0.5.0 - HTML rendering was moved to template with possibility to use custom templates
- 0.4.0 - added `breadcrumb_for` block tag
- 0.3.3 - fixed typo in 0.3.2
- 0.3.2 - added `breadcrumb_safe` tag



## CHAPTER 8

---

### Contributors

---

Author: Łukasz Mierzwa <l.mierzwa [at] gmail>

Contributors:

- Ewoud Kohl van Wijngaarden
- gnuwho
- Christian Dullweber
- Eric Davis (edavis)
- Guillaume DE BURE (gdebure)
- JeLoueMonCampingCar
- JP-Ellis
- Alexandre Macabies (zopieux)