django-activeurl Documentation

Release 0.1.12

hellysmile

Feb 18, 2018

Contents

Easy-to-use active URL highlighting for Django

CHAPTER 1

Features

- automatic highlighting of currently active <a> tags via CSS class
- automatic highlighting of parent <a> tags for menus
- removes boring / hardcoded stuff from your life!
- href="#" never causes highlighting for compatibility with techniques such as bootstrap nav.

CHAPTER 2

Usage

After loading the template library via

```
{% load activeurl %}
```

the following code snippet will be rendered like this if *request.full_path()* starts with */some_page/*:

```
{% activeurl %}
    <ul>
        <li> <!-- this <li> will render as <li class="active"> -->
            <a href="/some_page/">
                some_page
            </a>
        </li>
        <li>
            <a href="/another_page/">
                another_page
            </a>
        </li>
    </ul>
{% endactiveurl %}
```

Note: The content of `{% activeurl %}...{% endactiveurl %}` must have valid root tag (i.e. `<ul>` or `<div>`, etc) – otherwise an exception will be raised.

CHAPTER 3

Installation

Python 2.7+, 3.4+ are supported.

1. Install the *stable* version using `pip`:

```
pip install django-activeurl
```

   or install the *in-development* version using `pip`:

```
pip install -e git+git://github.com/hellysmile/django-activeurl#egg=django_
↪activeurl
```

2. In your `settings.py` add the following:

```
INSTALLED_APPS = (
    ...
    'django_activeurl',
    ...
)

TEMPLATE_CONTEXT_PROCESSORS = (
    ...
    'django.core.context_processors.request',
    ...
)
```

3. The *lxml* library is required and thus additional libraries need to be installed to build it:

   - Ubuntu:

```
sudo apt-get install libxml2 libxml2-dev libxslt-dev build-essential python-
↪dev
sudo ldconfig
```

   - Fedora:

```
sudo yum groupinstall 'Development Tools'
sudo yum install libxslt-devel libxml2 libxml2-devel python-devel
sudo ldconfig
```

- MacOS X:

```
brew install libxml2 libxslt
sudo update_dyld_shared_cache -force
```

- Windows: A pre-built *lxml* binary can be found here
- Clouds: There's a 99.99% chance that *lxml* will build out of the box.

CHAPTER 4

Options

4.1 menu =”yes|no” (default: “yes”)

Should hierarchical menus be supported? There are two different ways to declare an *active* status:

- the *starts-with* logic toggles the active state if `request.get_full_path()` starts with the contents of the `<a href=` attribute.
- the *equals* logic toggles the active state if `request.get_full_path()` is identical to the contents of the `<a href=` attribute.

You might want to use **starts-with logic** in hierarchical menus/submenus to not only highlight the current position but also every parent position. So, with `request.get_full_path()` being */menu/submenu* the following snippet will render accordingly:

```
{% activeurl menu="yes" parent_tag="div" %}
    <div>
        <div>  <!-- This will render as <div class="active"> -->
            <a href="/menu/">
                menu
            </a>
            <div>  <!-- This will also render as <div class="active"> -->
                <a href="/menu/submenu/">
                    submenu
                </a>
            </div>
        </div>
    </div>
{% endactiveurl %}
```

The **equals** logic works best for non-hierarchical menus where only those items should be highlighted whose `href`-attribute perfectly match `request.get_full_path()`:

```
{% activeurl menu="no" parent_tag="div" %}
    <div>
```

```
        <div>
            <a href="/menu/">
                menu
            </a>
        </div>
        <div>
            <a href="/menu/submenu/">
                submenu
            </a>
        </div>
    </div>
{% endactiveurl %}
```

4.2 ignore_params =”yes|no” (default: “no”)

`ignore_params` will ignore GET parameters of URLs, e.g. */accounts/login/* will match */accounts/login/?next=/accounts/signup/*.

4.3 parent_tag =”div|li|self|...” (default: “li”)

`parent_tag` defines that a parent element – and not the `<a>` tag itself – should be declared *active* when there's a match in URLs. When you need to change the CSS class of the `<a>` tag, just enter “self”.

4.4 css_class =”<string>” (default: “active”)

Defines what CSS class to add to an active element.

CHAPTER 5

Configuration

The default options can be set in `settings.py` as well:

```
ACTIVE_URL_KWARGS = {
    'css_class': 'active',
    'parent_tag': 'li',
    'menu': 'yes',
    'ignore_params': 'no'
}
ACTIVE_URL_CACHE = True
ACTIVE_URL_CACHE_TIMEOUT = 60 * 60 * 24  # 1 day
ACTIVE_URL_CACHE_PREFIX = 'django_activeurl'
```

By default *django-activeurl* will try to retrieve a previously rendered HTML node from Django's caching backend before active URLs are looked for and a new HTML tree is built. You can disable the cache with `ACTIVE_URL_CACHE = False`.

In addition, `ACTIVE_URL_CACHE_TIMEOUT` can be used to define a timeout for keys to expire. The default value is one day.

The last configuration option is `ACTIVE_URL_CACHE_PREFIX` (which is `django_activeurl` by default) and defines which name to use in Django's caching backend.

5.1 Tests

```
pip install tox
tox
```

CHAPTER 6

Jinja2

Vanilla Jinja2 configuration:

```
from jinja2 import Environment

from django_activeurl.ext.django_jinja import ActiveUrl


env = Environment(
    extensions=[ActiveUrl]
)
```

Options can be omitted:

```
{% activeurl css_class="active", menu="yes", parent_tag="li", ignore_params="no" %}
    <ul>
        <li>
            <a href="/page/">page</a>
        </li>
        <li>
            <a href="/other_page/">other_page</a>
        </li>
    </ul>
{% endactiveurl %}
```

If you're using django-jinja you need to load the `ActiveUrl` in `settings.py`.

Django 1.8+ Jinja2 environment loader example can be found in tests.

CHAPTER 7

Background

For building the HTML element tree *django-activeurl* uses lxml, which is one of the best HTML parsing tools around. More info and benchmarks can be found at habrahabr.ru (in russian). Note that there's no content rebuilding inside the template tag when no active URLs are found, so there's no impact on performance.

CHAPTER 8

Table of Contents

8.1 Changes

Here you can find the recent changes to django-activeurl

8.1.1 0.1.12 February 16, 2018

- Ignore href="#" to fix incompatibilities with bootstrap. This matches <= 0.1.9 behaviour.

8.1.2 0.1.11 October 10, 2017

- ignore_params now works with menu="no"

8.1.3 0.1.10 July 28, 2017

- Changelog started
- Added `ignore_params` for matching patterns with GET parameters.

  e.g. */path/* will match */path/?param=value*

  To enable this, add `ignore_params="yes"` to your `{% activeurl %}` tag:

```
{% activeurl ignore_params="yes" %}
```