latest

Release 0.4.2

January 23, 2017

Created by Stephen McDonald

A Django reusable app providing the `overextends` template tag, a drop-in replacement for Django's `extends` tag, which allows you to use circular template inheritance.

The primary use-case for `overextends` is to simultaneously override and extend templates from other reusable apps, in your own Django project.

CHAPTER 1

Example

Consider the following settings module and templates, with the apps `app1` and `app2` bundled in the project, for example's sake:

```
# settings.py
INSTALLED_APPS = (
    "app1",
    "app2",
    "overextends",
)
TEMPLATE_LOADERS = (
    "django.template.loaders.filesystem.Loader",
    "django.template.loaders.app_directories.Loader",
)
PROJECT_ROOT = os.path.dirname(os.path.abspath(__file__))
TEMPLATE_DIRS = (os.path.join(PROJECT_ROOT, "templates"),)

<!-- myproject/app1/templates/pages/page.html -->
<h1>Title</h1>
{% block main %}
<p>A paragraph in app1</p>
{% enblock %}
<footer>Copyright 2012</footer>

<!-- myproject/app2/templates/pages/page.html -->
{% overextends "pages/page.html" %}
{% block main %}
<p>A paragraph in app2, that wants to be on top of app1's main block</p>
{{ block.super }}
{% enblock %}

<!-- myproject/templates/pages/page.html -->
{% overextends "pages/page.html" %}
{% block main %}
{{ block.super }}
<p>A paragraph in the project's template directory, under the other main blocks</p>
{% enblock %}
```

The resulting HTML rendered when `pages/page.html` was loaded would be:

```
<h1>Title</h1>
<p>A paragraph in app2, that wants to be on top of app1's main block</p>
<p>A paragraph in app1</p>
<p>A paragraph in the project's template directory, under the other main blocks</p>
```

```
<footer>Copyright 2012</footer>
```

For a detailed analysis of why you would use this approach, how it works, and alternative approaches, read my initial blog post: Circular Template Inheritance for Django

CHAPTER 2

Installation

The easiest way to install django-overextends is directly from PyPi using pip by running the following command:

```
$ pip install -U django-overextends
```

Otherwise you can download django-overextends and install it directly from source:

```
$ python setup.py install
```

CHAPTER 3

Project Configuration

Once installed you can configure your project to use django-overextends by adding the `overextends` app to the `INSTALLED_APPS` in your project's `settings` module:

```
INSTALLED_APPS = (
    # ... other apps here ...
    'overextends',
)
```

For Django 1.9+ you must add overextends to the *builtins* key of your *TEMPLATES* setting:

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'APP_DIRS': True,
        'OPTIONS': {
            'builtins': ['overextends.templatetags.overextends_tags'],
        }
    },
]
```

Note that while the `overextends` tag is provided by the package `overextends.templatetags.overextends_tags`, it is unnecessary to use `{% load overextends_tags %}` in your templates. Like the `extends` tag, `overextends` must be the first tag in your template, so it is automatically added to Django's built-in template tags, removing the need to load its tag library in each template.