Django Globals Documentation

Release 0.3.1

Alexander Artemenko

Sep 09, 2018

Contents

1 Installation **3**

2 Configuration **5**

3 Usage **7**

3.1 Changelog 7

4 Indices and tables **9**

Django-globals is a very simple application, that allow you to define thread specific global variables.

It includes a middleware Global, which can be used to access to the current request and user, which is useful outside of a view when the "request" variable is not defined.

CHAPTER 1

Installation

Install using pip:

```
pip install django_globals
```

CHAPTER 2

Configuration

In your project's `settings.py`, add `django_globals.middleware.Global` to `MIDDLEWARE` (or `MIDDLEWARE_CLASSES` on Django < 1.10). eg:

```
MIDDLEWARE = (
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django_globals.middleware.Global',
)
```

CHAPTER 3

Usage

Now you can use `from django_globals import globals` and access to the `globals.request` and `globals.user` from anywhere.

3.1 Changelog

3.1.1 0.3.1

- Bugfix for 0.3.0 - handle AttributeError due to short-circuiting behaviour of old-style middleware

3.1.2 0.3.0

- Add support for new Django `MIDDLEWARE` setting introduced in Django 1.10.
- Invalidate global variables `request` and `response` at the end of the request/response cycle.
- Removed deprecated `django_globals.middleware.User`. Use `django_globals.middleware.Global` instead.

3.1.3 0.2.1

- A special code was added to send ReST version of readme to pypi.

3.1.4 0.2.0

- The middleware now save the user AND the request.
- The `django_globals.middleware.User` middleware is replaced by the `django_globals.middleware.Global` (but it's kept for retro-compatibility with a deprecation warning)

3.1.5 0.1.0

- First public release.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search