Django-Template-Debug Documentation

Release 0.3.3

Caleb Smith

Nov 17, 2017

Contents

A small collection of template tags for debugging and introspecting Django templates

Documentation

CHAPTER 1

Requirements

None, but the latest ipdb is highly recommended.

CHAPTER 2

Installation

django-template-debug is available on pypi, so the easiest way to install it is using pip:

```
pip install django-template-debug
```

CHAPTER 3

Setup

Add 'template_debug' to the INSTALLED_APPS iterable in your settings file. For example:

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

Add `TEMPLATE_DEBUG = True` to your local or development settings if it is not already set.

- Unless TEMPLATE_DEBUG is set to True, the django-template-debug templates will return an empty string without doing anything. This behavior prevents your application from calling set_trace() or print in a production environment if django-template-debug template tags are accidentally commited and deployed.

CHAPTER 4

Usage

Add {% load debug_tags %} in any Django template.

The available tags to use are {% set_trace %} {% variables %} {% attributes varname %} and {% details varname %}

See Example Usage docs for more details

CHAPTER 5

Developer Setup

Create a fresh virtualenv and install the test requirements:

```
mkvirtualenv template-debug
pip install -r requirements/test.txt
```

Use manage.py in the project directory along with the example.settings file for local testing.

To run unittests using the virtualenv's Python and Django, use the *runtests* script. To test all supported versions of Python and Django, run the unittests using tox.

5.1 Contents

CHAPTER 6

Indices and tables

- search