django-capture-tag Documentation

Release 1.0

Diederik van der Boor

April 20, 2016

Contents

1 Installation **3**

2 Syntax **5**

3 Example usage **7**

3.1 Notice 7

A micro-library to capture output in Django templates.

This can be useful for example to:

- Repeat page titles in web pages, e.g. for the `<title>` tag and breadcrumb.
- Repeat contents for Social Media tags.
- Reusing thumbnail output in multiple places.
- Fetch configuration data from extended templates.

CHAPTER 1

Installation

Install the module from PyPI:

```
pip install django-capture-tag
```

Add the package to `INSTALLED_APPS`:

```
INSTALLED_APPS += (
    'capture_tag',
)
```

Load the tag in your template:

```
{% load capture_tags %}
```

CHAPTER 2

Syntax

The following options are available:

```
{% capture %}...{% endcapture %}                    # output in {{ capture }}
{% capture silent %}...{% endcapture %}             # output in {{ capture }} only
{% capture as varname %}...{% endcapture %}         # output in {{ varname }}
{% capture as varname silent %}...{% endcapture %}  # output in {{ varname }} only
```

CHAPTER 3

Example usage

To capture Social Media tags:

```
{% load capture_tags %}

<head>
    ...

    {# Allow templates to override the page title/description #}
    <meta name="description" content="{% capture as meta_description %}{% block meta-description %}{
    <title>{% capture as meta_title %}{% block meta-title %}Untitled{% endblock %}{% endcapture %}</t

    {# display the same value as default, but allow templates to override it. #}
    <meta property="og:description" content="{% block og-description %}{{ meta_description }}{% endbl
    <meta name="twitter:title" content="{% block twitter-title %}{{ meta_title }}{% endblock %}" />
</head>
```

Take configuration from extended templates:

```
# base.html

{% load capture_tags %}

# read once
{% capture as home_url silent %}{% block home_url %}{% url 'app:index' %}{% endblock %}{% endcapture

# reuse twice.
<a href="{{ home_url }}" class="btn page-top">Back to home</a>
<a href="{{ home_url }}" class="btn page-bottom">Back to home</a>

# child.html
{% extends "base.html" %}

{% block home_url %}{% url 'user:profile' %}{% endblock %}
```

3.1 Notice

When a value is used only once, this package is not needed. In such case, simply place the `{% block .. %}` at the proper location where contents is replaced. All common Django template tags support the `as variable` syntax, such as `{% url 'app:index' as home_url %}` or `{% trans "Foo" as foo_label %}`.