django-tinycontent Documentation

Release 0.1.0

Dominic Rodger

July 21, 2015

Contents

django-tinycontent is a simple Django application for re-usable content blocks, much like django-boxes.

Installation is simple:

```
pip install django-tinycontent
```

Add `tinycontent` to your `INSTALLED_APPS`.

Usage in templates is simple:

```
{% load tinycontent_tags %}

{% tinycontent_simple 'content_name' %}
```

Or, to specify a value if a content block by the given name cannot be found, use:

```
{% load tinycontent_tags %}

{% tinycontent 'content_name' %}
This will be shown if no matching object is found.
{% endtinycontent %}
```

The name of the content block can also be a context variable, using both the simple and the complex variants.

Content blocks themselves can be added and edited using Django's admin interface. If a block with the name given in the template tag cannot be found, either nothing is rendered (if using `tinycontent_simple`), or the text between `tinycontent` and `endtinycontent` is rendered (if using the more complex variant).

CHAPTER 1

Available Settings

1.1 TINYCONTENT_FILTER

New in version 0.1.8.

Set this to a dotted path to a function to call to filter the content (for example, to convert Markdown to HTML). If the given path is invalid, any use of tinycontent tags will raise `ImproperlyConfigured`. If this setting is not provided, the content will be returned exactly as stored.

For example, if your project has a file called `utils.py`, you might have a function in it called `tinycontent_transform` that would look something like this:

```
def tinycontent_transform(content):
    return do_something_to(content)
```

To get the tinycontent templates to use that function, in your `settings.py` file, you'd write something like:

```
TINYCONTENT_FILTER = 'myproj.utils.tinycontent_transform'
```