Markdownify Documentation

Release 1.0.0

Erwin Matijsen

Jul 18, 2020

Table of contents

CHAPTER 1

Requirements

Django Markdownify requires Django (obviously), as well as Markdown and Bleach version 2 or higher. When installing Django Markdownify, dependencies will be installed automatically.

CHAPTER 2

Installation

Install Django Markdownify with pip:

`pip install django-markdownify`

Or add `django-markdownify` to your requirements.txt and run `pip install -r requirements.txt`

Finally add `markdownify` to your installed apps in `settings.py`:

```
INSTALLED_APPS = [
    ...
    'markdownify',
]
```

CHAPTER 3

Usage

Load the tag in your template:

```
{% load markdownify %}
```

Then you can change markdown to html as follows:

```
{{ 'text'|markdownify }}
```

Use Markdown in your template directly:

```
{% load markdownify %}
{{'Some *test* [link](#)'|markdownify }}
```

Or use the filter on a variable passed to the template via your views. For example:

```
#views.py
class MarkDown(TemplateView):
    template_name = 'index.html'

    def get_context_data(self, **kwargs):
        markdowntext = open(os.path.join(os.path.dirname(__file__), 'templates/test.md
↪')).read()

        context = super(MarkDown, self).get_context_data(**kwargs)
        context['markdowntext'] = markdowntext

        return context

#index.html
{% load markdownify %}
{{ markdowntext|markdownify }}
```

CHAPTER 4

Settings

You can change the behavior of Markdownify by adding them to your `settings.py`. All settings are optional and will fall back to default behavior if not specified.

4.1 Whitelist tags

Add whitelisted tags with `MARKDOWNIFY_WHITELIST_TAGS = []` For example:

```
MARKDOWNIFY_WHITELIST_TAGS = [
    'a',
    'abbr',
    'acronym',
    'b',
    'blockquote',
    'em',
    'i',
    'li',
    'ol',
    'p',
    'strong',
    'ul'
]
```

`MARKDOWNIFY_WHITELIST_TAGS` defaults to bleach.sanitizer.ALLOWED_TAGS

4.2 Whitelist attributes

Add whitelisted attributes with `MARKDOWNIFY_WHITELIST_ATTRS = []` For example:

```
MARKDOWNIFY_WHITELIST_ATTRS = [
    'href',
    'src',
    'alt',
]
```

MARKDOWNIFY_WHITELIST_ATTRS defaults to bleach.sanitizer.ALLOWED_ATTRIBUTES

4.3 Whitelist styles

Add whitelisted styles with MARKDOWNIFY_WHITELIST_STYLES = [] For example:

```
MARKDOWNIFY_WHITELIST_STYLES = [
    'color',
    'font-weight',
]
```

MARKDOWNIFY_WHITELIST_STYLES defaults to bleach.sanitizer.ALLOWED_STYLES (Note that it's an empty list)

4.4 Whitelist protocols

Add whitelisted protocols with MARKDOWNIFY_WHITELIST_PROTOCOLS = [] For example:

```
MARKDOWNIFY_WHITELIST_PROTOCOLS = [
    'http',
    'https',
]
```

MARKDOWNIFY_WHITELIST_PROTOCOLS defaults to bleach.sanitizer.ALLOWED_PROTOCOLS

4.5 Enable Markdown Extensions

Python-Markdown is extensible with extensions. To enable one or more extensions, add MARKDOWNIFY_MARKDOWN_EXTENSIONS to your settings.py. For example:

```
MARKDOWNIFY_MARKDOWN_EXTENSIONS = ['markdown.extensions.fenced_code',
                                   'markdown.extensions.extra', ]
```

MARKDOWNIFY_MARKDOWN_EXTENSIONS defaults to an empty list (so no extensions are used). To read more about extensions and see the list of official supported extensions, go to the markdown documentation.

4.6 Strip markup

Choose if you want to strip or escape tags that aren't allowed. MARKDOWNIFY_STRIP = True (default) strips the tags. MARKDOWNIFY_STRIP = False escapes them.

4.7 Disable sanitation (bleach)

If you just want to markdownify your text, not sanitize it, set `MARKDOWNIFY_BLEACH = False`. Defaults to `True`.

4.8 Linkify text

Use `MARKDOWNIFY_LINKIFY_TEXT` to choose if you automatically want your links to be rendered to hyperlinks. Defaults to `MARKDOWNIFY_LINKIFY_TEXT = True`. If `True`, links will be linkified but emailaddresses won't.

Use the following settings to change the linkify behavior:

4.8.1 Linkify email

Set `MARKDOWNIFY_LINKIFY_PARSE_EMAIL` to `True` or `False` to automatically linkify emailaddresses found in your text. Defaults to `False`.

4.8.2 Set callbacks

Set `MARKDOWNIFY_LINKIFY_CALLBACKS` to use callbacks to modify your links, for example setting a title attribute to all your links.:

```
def set_title(attrs, new=False):
    attrs[(None, u'title')] = u'link in user text'
    return attrs

# settings.py
MARKDOWNIFY_LINKIFY_CALLBACKS = [set_title, ]
```

`MARKDOWNIFY_LINKIFY_CALLBACKS` defaults to `None`, so no callbacks are used. See the bleach documentation for more examples.

4.8.3 Skip tags

Add tags with `MARKDOWNIFY_LINKIFY_SKIP_TAGS = []` to skip linkifying links within those tags, for example `<pre>` blocks. For example:

```
MARKDOWNIFY_LINKIFY_SKIP_TAGS = ['pre', 'code', ]
```

CHAPTER 5

Tests

Django Markdownify comes with tests to check if settings and defaults produce the expected output. To run the tests, make sure Django Markdownify is *installed*. Then go to your project where Django Markdownify is installed, and run the tests.

```
>>> cd /path/to/your/project
>>> python manage.py test markdownify
```

CHAPTER 6

Django Markdownify - A Django Markdown filter

Django Markdownify is a template filter to convert Markdown to HTML in Django. Markdown is converted to HTML and sanitized.

Example:

```
{% load markdownify %}
{{'Some *test* [link](#)'|markdownify }}
```

Is transformed to:

```
<p>
  Some <em>test</em> <a href="#">link</a>
</p>
```

The code can be found on github.