
djangocms-page-meta Documentation

Release 0.8.0.dev1

Iacopo Spalletti

Dec 30, 2017

Contents

1	djangoCMS-page-meta	3
1.1	Quickstart	3
1.2	Documentation	4
2	Usage	5
2.1	Assigning meta information to pages / titles	5
2.2	Rendering meta information	5
2.3	Supported attributes	6
2.4	Generic meta	7
2.5	Templatetags	8
3	djangoCMS_page_meta.utils	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Credits	15
5.1	Development Lead	15
5.2	Contributors	15
6	History	17
6.1	0.8.0 (unreleased)	17
6.2	0.7.0 (2016-12-04)	17
6.3	0.6.3 (2016-10-22)	17
6.4	0.6.2 (2016-08-01)	17
6.5	0.6.1 (2016-06-20)	18
6.6	0.6 (2016-06-05)	18
6.7	0.5.11 (2016-03-17)	18
6.8	0.5.10 (2016-03-03)	18
6.9	0.5.9 (2016-02-25)	18
6.10	0.5.8 (2016-02-23)	18
6.11	0.5.7 (2016-01-09)	18
6.12	0.5.6 (2015-10-03)	18
6.13	0.5.5 (2015-08-21)	19

6.14	0.5.4 (2015-08-08)	19
6.15	0.5.3 (2015-06-29)	19
6.16	0.5.2 (2015-06-28)	19
6.17	0.5.1 (2015-06-06)	19
6.18	0.5 (2015-06-06)	19
6.19	0.4.1 (2014-08-10)	19
6.20	0.4.0 (2014-04-19)	20
6.21	0.3.3 (2014-03-23)	20
6.22	0.3.2 (2014-03-07)	20
6.23	0.3.1 (2014-02-13)	20
6.24	0.3.0 (2014-02-02)	20
6.25	0.2.1 (2014-01-26)	20
6.26	0.2.0 (2014-01-22)	20
6.27	0.1.0 (2014-01-21)	20
7	Indices and tables	21
	Python Module Index	23

Contents:

djangoCMS-page-meta

Meta tag information for django CMS 3 pages

Python: 2.7, 3.4, 3.5, 3.6

Django: 1.8 to 1.11

django CMS: 3.4 (and develop/3.5)

Warning: Since version 0.7, the support for Python 2.6, Python 3.3, Django<1.8 and django CMS<3.2 has been dropped

1.1 Quickstart

1. A working django CMS environment is required for djangoCMS-page-meta to work. Refer to [django CMS documentation](#) for how to install and configure django CMS.
2. Install djangoCMS-page-meta:

```
$ pip install djangoCMS-page-meta
```

or from the repository:

```
pip install -e git+https://github.com/nephila/djangoCMS-page-meta#egg=djangoCMS-  
↪page-meta
```

3. Then add it to INSTALLED_APPS along with its dependencies:

```
'filer',  
'meta',  
'easy_thumbnails',  
'djangoCMS_page_meta',
```

4. Synchronize the database with syncdb:

```
$ python manage.py syncdb
```

or migrate:

```
$ python manage.py migrate
```

5. Configuration:

There is no special configuration for djangoCMS-page-meta, but make sure you set the necessary configuration for [django-meta](#), especially the [template setup](#) section

6. That's all!

1.1.1 Dependencies

- [django-filer](#) `>= 1.2`
- [django-meta](#) `>= 1.3`

1.2 Documentation

For package documentation see <https://djangoCMS-page-meta.readthedocs.io/>.

2.1 Assigning meta information to pages / titles

Meta information can be assigned from the admin interface or the toolbar.

In the toolbar you will find a `Meta-information` submenu in the `Page` menu, with two or more sub-items:

- Common: it allows to edit page-wide (language independent) meta information;
- One entry per active language to edit language dependent information.

2.2 Rendering meta information

To render provided meta information you must add these lines in the main template:

```
{% load page_meta_tags %}
{% page_meta request.current_page as meta %}

[...]
<!-- This must be in the head -->
{% include 'djangocms_page_meta/meta.html' with meta=meta %}
```

When using Google+ you must add the `type` attribute to the `body` or `html` tag:

```
<html {% googleplus_html_scope meta.gplus_type %}>
```

or:

```
<body {% googleplus_html_scope meta.gplus_type %}>
```

2.3 Supported attributes

djangocms-page-meta currently offers partial support for [OpenGraph](#), [Twitter Cards](#) and [Google+ Snippets](#). As a generic application djangocms-page-meta cannot cover every use case while still being useful to most people.

2.3.1 Generic HTML

- description: HTML meta description of the page
- keywords: HTML meta keywords

2.3.2 OpenGraph

The following properties are supported:

- og:title
- og:url
- og:description
- og:image
- og:type
- og:site_name
- og:locale
- article:author:url
- article:author:first_name
- article:author:last_name
- article:published_time
- article:modified_time
- article:expiration_time
- article:publisher
- article:tag
- fb:app_id
- fb:profile_id
- fb:pages

See [Facebook OpenGraph documentation](#) for more information about each property.

2.3.3 Twitter Cards

The following properties are supported:

- twitter:domain
- twitter:card
- twitter:title

- twitter:url
- twitter:description
- twitter:image
- twitter:creator
- twitter:site_name

See [Twitter documentation](#) for more information about each property.

2.3.4 Google+ Snippets

Support for [Google+ Snippets](#) is very basic, and limited to the `<head>` part of the page. As Google+ Snippets uses [Schema.org microdata](#) to add meta-information to the markup, you might need to further customize the markup according to you specific content.

As of now support is limited to the the following data:

- rel=author, via `link rel="author"` in the `<head>`
- name
- image
- datePublished
- dateModified
- url
- description
- image
- type (i.e. itemscope), appended to `<html>` or `<body>` tag

Currently all the accepted values for **type** are provided as valid choices; not all of them are actually sensible values for CMS pages and `djangoCMS-page-meta` offers limited support for the attributes required by some accepted types.

Article or Blog type should be sensible for most use cases.

2.4 Generic meta

If different metas are needed, a generic model exists that allows to add custom metas.

Both `PageMeta` and `TitleMeta` provides an inline model that allows to define custom metas; model provides three fields:

- attribute: meta attribute
- name: name of the meta
- value: value of the meta

Each inline will be rendered as:

```
<meta {{ attribute }}="{{ {{ name }}" content="{{ {{ value }}" />
```

2.5 Templatetags

2.5.1 page_meta

`page_meta` templatetags extract information from the given page to a context variable that can be passed to the included template for rendering.

Arguments:

- `page`: a page instance (typically current page);
- `varname`: the name of the context variable to save data to.

djangocms_page_meta.utils

Warning: While functions in this module can be used in code to work with the data, they are by no means to be considered as a stable API, so they may change in future release.

`djangocms_page_meta.utils.get_cache_key` (*page*, *language*)
Create the cache key for the current page and language

`djangocms_page_meta.utils.get_page_meta` (*page*, *language*)
Retrieves all the meta information for the page in the given language

Parameters

- **page** – a Page instance
- **lang** – a language code

Returns Meta instance

Type object

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/nephila/djangocms-page-meta/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Translations

Any translation is welcome. Please use Transifex <https://www.transifex.com/projects/p/djangocms-page-meta/> to help with translations.

4.1.5 Write Documentation

djangocms-page-meta could always use more documentation, whether as part of the official djangocms-page-meta docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.6 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/nephila/djangocms-page-meta/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *djangocms-page-meta* for local development.

1. Fork the *djangocms-page-meta* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/djangocms-page-meta.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv djangocms-page-meta
$ cd djangocms-page-meta/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 djangocms_page_meta tests
$ python setup.py test
$ tox
```

To get flake8 and tox install them into your virtualenv:

```
$ pip install -r requirements-test.txt
```

6. Commit your changes and push your branch to GitHub:

```
$ git add .  
$ git commit -m "Your detailed description of your changes."  
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/nephila/djangocms-page-meta/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python runtests.py
```


5.1 Development Lead

- Iacopo Spalletti <i.spalletti@nephila.it>

5.2 Contributors

- Adam Chainz
- David D Lowe
- Federico Frenguelli
- Grant Eagon
- Leonardo Cavallucci
- Marco Federighi
- Mario Colombo
- nasty
- Ryan Stalbow
- Sergei Maertens
- Tadas Dailyda
- Venelin Stoykov
- Zan Anderle

6.1 0.8.0 (unreleased)

- Add Django 1.11 support
- Drop django CMS 3.2, 3.3
- Add support for filer canonical URL
- Do not fail if request is not in context

6.2 0.7.0 (2016-12-04)

- Drop Django 1.7 and below
- Drop django CMS 3.1 and below
- Add Django 1.10 support

6.3 0.6.3 (2016-10-22)

- Added support for django CMS 3.4

6.4 0.6.2 (2016-08-01)

- Conditionally show page description field if not empty

6.5 0.6.1 (2016-06-20)

- Drop unique index on generic meta attributes

6.6 0.6 (2016-06-05)

- Let settings be used as defaults
- Added support for fb_pages attribute
- Added support for generic meta attributes

6.7 0.5.11 (2016-03-17)

- Make object type not required

6.8 0.5.10 (2016-03-03)

- Fix handling Google+ author URL

6.9 0.5.9 (2016-02-25)

- Fix issue with Django 1.9

6.10 0.5.8 (2016-02-23)

- Improve Aldryn support
- Catch AttributeError in get_page_meta

6.11 0.5.7 (2016-01-09)

- Fix handling of pages in non existing languages
- Add support for Aldryn environment

6.12 0.5.6 (2015-10-03)

- Fix error when retrieving tags from djangocms-page-tags
- Add support for django CMS 3.2
- Add support for Django 1.8

6.13 0.5.5 (2015-08-21)

- Different approach at fix error when page is None
- Better fallback mechanism
- Update i18n

6.14 0.5.4 (2015-08-08)

- Fix error when page is None
- Code cleanup
- Add pep8/isort to tests
- Add missing migration

6.15 0.5.3 (2015-06-29)

- Fix Django 1.7 migration
- Update testing configuration

6.16 0.5.2 (2015-06-28)

- Compatibility with django CMS 3.1.1

6.17 0.5.1 (2015-06-06)

- Make south migrations independent from djangoCMS-helper

6.18 0.5 (2015-06-06)

- Add support for custom user model
- Fix the length of SEO fields
- Use djangoCMS-helper for tests
- Add support for Django 1.7
- Add support for django CMS 3.1

6.19 0.4.1 (2014-08-10)

- Minor fixes

6.20 0.4.0 (2014-04-19)

- django CMS 3 final release support
- Moved some code to external django-meta-mixin application

6.21 0.3.3 (2014-03-23)

- Better support for sn-dependent descriptions

6.22 0.3.2 (2014-03-07)

- Add different descriptions for each supported social network

6.23 0.3.1 (2014-02-13)

- Fix permission name

6.24 0.3.0 (2014-02-02)

- Fix for django CMS 3 develop update

6.25 0.2.1 (2014-01-26)

- Vendorizing django-meta compatibility templatetags

6.26 0.2.0 (2014-01-22)

- Put toolbar menu items in a submenu

6.27 0.1.0 (2014-01-21)

- Experimental release.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

d

`djangoCMS_page_meta.utils`, 9

D

`djangoCMS_page_meta.utils` (module), 9

G

`get_cache_key()` (in module `djangoCMS_page_meta.utils`),
9

`get_page_meta()` (in module `djangoCMS_page_meta.utils`), 9