
djangocms-page-meta Documentation

Release 1.0.0

Iacopo Spalletti

Nov 18, 2020

Contents

1	djangoCMS-page-meta	3
1.1	Quickstart	3
1.2	Documentation	4
2	Configuration	5
2.1	Settings	5
2.2	django-meta configuration	5
3	Usage	7
3.1	Assigning meta information to pages / titles	7
3.2	Rendering meta information	7
3.3	Supported attributes	8
3.4	Generic meta	10
3.5	Templatetags	10
4	djangoCMS_page_meta.utils	11
5	Contributing	13
5.1	Types of Contributions	13
6	Credits	17
6.1	Development Lead	17
6.2	Contributors	17
7	History	19
7.1	1.0.0 (2020-11-18)	19
7.2	0.10.0 (2019-12-27)	19
7.3	0.9.0 (2019-08-22)	19
7.4	0.8.5 (2019-03-22)	20
7.5	0.8.4 (2019-03-11)	20
7.6	0.8.3 (2018-04-07)	20
7.7	0.8.2 (2018-02-22)	20
7.8	0.8.1 (2018-02-17)	20
7.9	0.8.0 (2018-01-14)	20
7.10	0.7.0 (2016-12-04)	21
7.11	0.6.3 (2016-10-22)	21
7.12	0.6.2 (2016-08-01)	21

7.13	0.6.1 (2016-06-20)	21
7.14	0.6 (2016-06-05)	21
7.15	0.5.11 (2016-03-17)	21
7.16	0.5.10 (2016-03-03)	21
7.17	0.5.9 (2016-02-25)	21
7.18	0.5.8 (2016-02-23)	22
7.19	0.5.7 (2016-01-09)	22
7.20	0.5.6 (2015-10-03)	22
7.21	0.5.5 (2015-08-21)	22
7.22	0.5.4 (2015-08-08)	22
7.23	0.5.3 (2015-06-29)	22
7.24	0.5.2 (2015-06-28)	22
7.25	0.5.1 (2015-06-06)	23
7.26	0.5 (2015-06-06)	23
7.27	0.4.1 (2014-08-10)	23
7.28	0.4.0 (2014-04-19)	23
7.29	0.3.3 (2014-03-23)	23
7.30	0.3.2 (2014-03-07)	23
7.31	0.3.1 (2014-02-13)	23
7.32	0.3.0 (2014-02-02)	23
7.33	0.2.1 (2014-01-26)	24
7.34	0.2.0 (2014-01-22)	24
7.35	0.1.0 (2014-01-21)	24
8	Indices and tables	25
	Python Module Index	27
	Index	29

Contents:

Meta tag information for django CMS 3 pages

Python: 3.6, 3.7, 3.8

Django: 2.2, 3.0, 3.1

django CMS: 3.7, 3.8

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:

See [usage](#) and [configuration](#) section in the documentation.

6. That's all!

Note: Enabling this will **hide** django CMS own **Meta description** field to keep all the meta information in the same part of the interface. If the django CMS field is set, it will still be shown (and used by djangoCMS-page-meta).

1.1.1 Dependencies

- django-filer
- django-meta

1.2 Documentation

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

2.1 Settings

2.1.1 PAGE_META_DESCRIPTION_LENGTH

Set the max length of the HTML meta description field. Default is 320.

2.1.2 PAGE_META_TWITTER_DESCRIPTION_LENGTH

Set the max length of the Twitter card description field. Default is 280.

2.2 django-meta configuration

django-cms-page-meta needs a working [django-meta](#) configuration, especially the [template setup](#) section.

Please check [django-meta configuration](#) for complete settings details.

3.1 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.

3.2 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 page_meta %}

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

When using microdata from [Schema.org](https://schema.org/docs/gs.html#microdata_how), you must add the `type` attribute to the body or html tag:

```
{% autoescape off %}
<html {% schemaorg_html_scope page_meta.schemaorg_type %}>
{% endautoescape %}
```

or:

```
{% autoescape off %}
<body {% schemaorg_html_scope page_meta.schemaorg_type %}>
{% endautoescape %}
```

Don't forget to load meta in your template!

```
{% load cms_tags menu_tags sekizai_tags page_meta_tags meta %}
```

3.3 Supported attributes

django-cms-page-meta currently offers partial support for OpenGraph, Twitter Cards and Schema.org microdata. As a generic application django-cms-page-meta cannot cover every use case while still being useful to most people.

3.3.1 Generic HTML

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

Note: Enabling this will **hide** django CMS own **Meta description** field to keep all the meta information in the same part of the interface. If the django CMS field is set, it will still be shown (and used by django-cms-page-meta).

3.3.2 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.

3.3.3 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.

3.3.4 Schema.org microdata

Support for [Schema.org microdata](#) is very basic, and limited to the `<html>` & `<body>` tags. 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.

3.4 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 }}" />
```

3.5 Templatetags

3.5.1 `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:

5.1 Types of Contributions

5.1.1 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.

5.1.2 Fix Bugs

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

5.1.3 Implement Features

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

5.1.4 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.

5.1.5 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 :)

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

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:

```
$ tox
```

To get tox, pip install it into your virtualenv.

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.

Development tips

This project allows you to use `pre-commit` to ensure an easy compliance to the project code styles.

If you want to use it, install it globally (for example with `pip3 install --user precommit`, but check [installation instruction](#). When first cloning the project ensure you install the git hooks by running `pre-commit install`.

From now on every commit will be checked against our code style.

Check also the available tox environments with `tox -l`: the ones not marked with a python version number are tools to help you work on the project buy checking / formatting code style, running docs etc.

Testing tips

You can test your project using any specific combination of python, django and django cms.

For example `tox -epy3.7-django30-cms37` runs the tests on python 3.7, Django 3.0 and django CMS 3.7.

5.1.6 Pull Request Guidelines

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

1. Pull request must be named with the following naming scheme:

`<type>/(<optional-task-type>-) <number>-description`

See below for available types.

2. The pull request should include tests.
3. If the pull request adds functionality, the docs should be updated. Documentation must be added in `docs` directory, and must include usage information for the end user. In case of public API method, add extended docstrings with full parameters description and usage example.
4. Add a changes file in `changes` directory describing the contribution in one line. It will be added automatically to the history file upon release. File must be named as `<issue-number>.<type>` with type being:
 - `.feature`: For new features.
 - `.bugfix`: For bug fixes.
 - `.doc`: For documentation improvement.
 - `.removal`: For deprecation or removal of public API.
 - `.misc`: For general issues.

Check [towncrier](#) documentation for more details.

5. The pull request should work for all python / django / django CMS versions declared in `tox.ini`. Check the CI and make sure that the tests pass for all supported versions.

5.1.7 Release a version

1. Update authors file
2. Merge `develop` on `master` branch
3. Bump release via task: `inv tag-release (major|minor|patch)`

4. Update changelog via towncrier: `towncrier --yes`
5. Commit changelog with `git commit --amend` to merge with `bumpversion commit`
6. Create tag `git tag <version>`
7. Push tag to github
8. Publish the release from the tags page
9. If pipeline succeeds, push `master`
10. Merge `master` back on `develop`
11. Bump development version via task: `inv tag-dev -l (major|minor|patch)`
12. Push `develop`

6.1 Development Lead

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

6.2 Contributors

- Adam Chainz
- Balazs Szele
- Christian Strappazzon
- David D Lowe
- Federico Frenguelli
- Grant Eagon
- Ian Epperson
- Kevin Cai
- Leonardo Cavallucci
- Marco Federighi
- Mario Colombo
- Mark Walker
- Matas Dailyda
- nastya
- Ryan Stalbow
- Sergei Maertens

- Tadas Dailyda
- Vadim Sikora
- Venelin Stoykov
- Vladimir Kuvandjiev
- Zan Anderle

7.1 1.0.0 (2020-11-18)

7.1.1 Features

- Update tooling and drop Python 2 / Django < 2.2 compatibility (#135)
- Add support for django CMS 3.8 / Django 3.1
- Add support for django-meta 2.0

7.1.2 Bugfixes

- Fix setup.cfg and pyproject.toml package name (#138)

7.2 0.10.0 (2019-12-27)

- Move from.djangocms-helper to django-app-helper
- Improve toolbar performance

7.3 0.9.0 (2019-08-22)

- Add support for django CMS 3.7 / Django 2.2
- Remove support for Django < 1.11 / Python 3.4

7.4 0.8.5 (2019-03-22)

- Fix error in migration missing on_delete argument

7.5 0.8.4 (2019-03-11)

- Add Django 2.0 and Django 2.1 compatibility
- Add django CMS 3.6 compatibility
- Fixes a crash on creating new pagetype with djangoCMS 3.5.2

7.6 0.8.3 (2018-04-07)

- Remove description when creating a CMS page
- Provide page dates meta if PageMeta does not exist
- Improve compatibility with Django 3.5

7.7 0.8.2 (2018-02-22)

- Change method of patching PageAdmin for better compatibility with other addons

7.8 0.8.1 (2018-02-17)

- Fix duplicate GenericMetaAttribute on publish
- Fix error on Divio cloud when description fields length is not set

7.9 0.8.0 (2018-01-14)

- Add Django 1.11 support
- Add django CMS 3.5 support
- Drop django CMS 3.2, 3.3
- Add support for filer canonical URL
- Do not fail if request is not in context
- Make HTML and twitter description fields length configurable

7.10 0.7.0 (2016-12-04)

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

7.11 0.6.3 (2016-10-22)

- Added support for django CMS 3.4

7.12 0.6.2 (2016-08-01)

- Conditionally show page description field if not empty

7.13 0.6.1 (2016-06-20)

- Drop unique index on generic meta attributes

7.14 0.6 (2016-06-05)

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

7.15 0.5.11 (2016-03-17)

- Make object type not required

7.16 0.5.10 (2016-03-03)

- Fix handling Google+ author URL

7.17 0.5.9 (2016-02-25)

- Fix issue with Django 1.9

7.18 0.5.8 (2016-02-23)

- Improve Aldryn support
- Catch AttributeError in get_page_meta

7.19 0.5.7 (2016-01-09)

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

7.20 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

7.21 0.5.5 (2015-08-21)

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

7.22 0.5.4 (2015-08-08)

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

7.23 0.5.3 (2015-06-29)

- Fix Django 1.7 migration
- Update testing configuration

7.24 0.5.2 (2015-06-28)

- Compatibility with django CMS 3.1.1

7.25 0.5.1 (2015-06-06)

- Make south migrations independent from djangocms-helper

7.26 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

7.27 0.4.1 (2014-08-10)

- Minor fixes

7.28 0.4.0 (2014-04-19)

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

7.29 0.3.3 (2014-03-23)

- Better support for sn-dependent descriptions

7.30 0.3.2 (2014-03-07)

- Add different descriptions for each supported social network

7.31 0.3.1 (2014-02-13)

- Fix permission name

7.32 0.3.0 (2014-02-02)

- Fix for django CMS 3 develop update

7.33 0.2.1 (2014-01-26)

- Vendorizing django-meta compatibility templatetags

7.34 0.2.0 (2014-01-22)

- Put toolbar menu items in a submenu

7.35 0.1.0 (2014-01-21)

- Experimental release.

CHAPTER 8

Indices and tables

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

d

`djangoCMS_page_meta.utils`, 11

D

`djangoCMS_page_meta.utils` (*module*), 11

G

`get_cache_key()` (*in module djangoCMS_page_meta.utils*), 11

`get_page_meta()` (*in module djangoCMS_page_meta.utils*), 11