
djangocms-highlightjs Documentation

Release 0.1.0

Iacopo Spalletti

February 02, 2014

Contents:

djangoCMS-highlightjs

highlight.js plugin for django CMS 3.0

1.1 Documentation

The full documentation is at <http://djangoCMS-highlightjs.rtd.org>.

1.2 Quickstart

1. Install djangoCMS-highlightjs:

```
pip install djangoCMS-highlightjs
```

2. Add to INSTALLED_APPS:

```
'djangoCMS_highlightjs',
```

3. Update the database schema:

```
$ python manage migrate djangoCMS_highlightjs
```

4. Add “**highlight.js code**” plugin to your placeholders

1.3 Features

- Use [highlight.js](#) to do syntax highlighting on provided code

2.1 Supported languages

- Apache
- Bash
- C#
- C++
- CSS
- CoffeeScript
- Diff
- Django
- HTML, XML
- HTTP
- Ini
- JSON
- Java
- JavaScript
- Makefile
- Markdown
- Nginx
- Objective C
- PHP
- Perl
- Python
- Ruby
- SCSS
- SQL

You can add more languages by compiling your own version of **highlight.js** at <http://highlightjs.org/download/> and put it in a `static/djangoCMS_highlight/js` directory in an application loaded **before** `djangoCMS_highlight`.

2.2 Themes

All default **highlight.js** themes are provided. You can write your own, put the relevant css file in `static/djangoCMS_highlight/css` directory in any application, add `HIGHLIGHT_THEME` in the project settings and update it to include your theme.

Contributing

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

3.1 Types of Contributions

3.1.1 Report Bugs

Report bugs at <https://github.com/nephila/djangocms-highlightjs/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.

3.1.2 Fix Bugs

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

3.1.3 Implement Features

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

3.1.4 Translations

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

3.1.5 Write Documentation

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

3.1.6 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/nephila/djangocms-highlightjs/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 :)

3.2 Get Started!

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

1. Fork the *djangocms-highlightjs* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/djangocms-highlightjs.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_highlightjs
$ cd djangocms-highlightjs/
$ 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_highlightjs tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them 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.

3.3 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-highlightjs/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

```
$ python runtests.py
```

Credits

4.1 Development Lead

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

4.2 Contributors

None yet. Why not be the first?

History

5.1 0.1.0 (2014-02-02)

- First release on PyPI.