django-jsnlog Documentation

Release 0.2.0

Tobias Funke

Apr 14, 2021

CONTENTS

1 django-jsnlog **3**
- 1.1 Documentation . . . 3
- 1.2 Quickstart . . . 3
- 1.3 Credits . . . 4

2 Installation **5**

3 Usage **7**
- 3.1 AppConfig . . . 7
- 3.2 URLs and Views . . . 7
- 3.3 Template . . . 8
- 3.4 Logging . . . 8

4 Contributing **9**
- 4.1 Types of Contributions . . . 9
- 4.2 Get Started! . . . 10
- 4.3 Pull Request Guidelines . . . 11
- 4.4 Tips . . . 11

5 Credits **13**
- 5.1 Development Lead . . . 13
- 5.2 Contributors . . . 13

6 History **15**
- 6.1 0.1.0 (2021-04-14) . . . 15

Contents:

CHAPTER

ONE

DJANGO-JSNLOG

A django integration for client side logging for javascript with JSNLog https://jsnlog.com/.

1.1 Documentation

The full documentation is at https://django-jsnlog.readthedocs.io.

1.2 Quickstart

Install django-jsnlog:

```
pip install django-jsnlog
```

Add it to your *INSTALLED_APPS*:

```
INSTALLED_APPS = (
    ...
    'jsnlog',
    ...
)
```

Add django-jsnlog's URL patterns:

```
urlpatterns += [
    path(r'jsnlog.logger', include('jsnlog.urls')),
]
```

Add django-jsnlog's javascript files to your template:

```
...
<script type="text/javascript" src="{% static 'jsnlog/js/jsnlog.min.js' %}"></script>
<script type="text/javascript" src="{% static 'jsnlog/js/django-jsnlog.js' %}"></
↪script>
...
```

1.3 Credits

Tools used in rendering this package:

- Cookiecutter
- cookiecutter-djangopackage

CHAPTER

TWO

INSTALLATION

At the command line:

```
$ easy_install django-jsnlog
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-jsnlog
$ pip install django-jsnlog
```

CHAPTER

THREE

USAGE

3.1 AppConfig

To use django-jsnlog in a project, add it to your *INSTALLED_APPS*:

```
# settings.py
INSTALLED_APPS = (
    ...
    'jsnlog',
    ...
)
```

or use your own AppConfig like this:

```
# settings.py
INSTALLED_APPS = (
    ...
    'myApp.apps.MyJSNLogConfig',
    ...
)

# myApp/apps.py
from jsnlog.apps import DefaultJSNLogConfig

class MyJSNLogConfig(DefaultJSNLogConfig):
    ...
```

3.2 URLs and Views

Add django-jsnlog's URL patterns:

```
urlpatterns += [
    path(r'jsnlog.logger', include('jsnlog.urls')),
]
```

or implement your own LogView by inheriting from JSNLogView like this:

```
# views.py
@method_decorator(login_required, name='dispatch')
class MyJSNLogView(JSNLogView):
    pass
```

(continues on next page)

(continued from previous page)

```
# urls.py
urlpatterns += [
    path(r'jsnlog.logger', MyJSNLogView.as_view()),
]
```

3.3 Template

Add django-jsnlog's javascript files to your template:

```
<script type="text/javascript" src="{% static 'jsnlog/js/jsnlog.min.js' %}"></script>
<script type="text/javascript" src="{% static 'jsnlog/js/django-jsnlog.js' %}"></
↪script>
```

3.4 Logging

Add django-jsnlog's logger to your project SETTINGS like this:

```
LOGGING = {
    ...
    'loggers': {
        ...
        # JSNLog logs (client js errors)
        'jsnlog': {
            'handlers': ['console'],
            'level': 'DEBUG',
            'propagate': True,
        },
        ...
    }
}
```

CHAPTER

FOUR

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:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/oruehenbeck/django-jsnlog/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 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/oruehenbeck/django-jsnlog/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 *django-jsnlog* for local development.

1. Fork the *django-jsnlog* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/django-jsnlog.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 django-jsnlog
$ cd django-jsnlog/
$ 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:

```
$ flake8 jsnlog tests
```

   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.

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/oruehenbeck/django-jsnlog/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

CHAPTER

FIVE

CREDITS

5.1 Development Lead

- Tobias Funke <tfunke@bona-fide.it>

5.2 Contributors

None yet. Why not be the first?

CHAPTER

SIX

HISTORY

6.1 0.1.0 (2021-04-14)

- First release on PyPI.