
Create issues in Github for internal server errors or exceptions in background tasks. Documentation

Release 0.1.0

Vedarth Kulkarni

April 03, 2015

1	Exreporter: Report Internal Server Errors	3
1.1	Features	3
1.2	Usage	3
1.3	Documentation	3
1.4	TODOS	3
2	Installation	5
3	Usage	7
3.1	Exreporter in Django Project	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
7	0.1.0 (2014-12-11)	17
8	0.1.1 (2014-12-15)	19
9	0.1.2 (2015-04-04)	21
10	Indices and tables	23

Contents:

Exreporter: Report Internal Server Errors

Exreporter is an MIT Licensed library, written in Python to report internal server errors and exceptions in background applications to issue trackers such as Github.

This will help in reducing the dependency on error emails for internal server error notifications and to manage them in a single, right place. Reported issues contain everything a developer needs, debugging got a whole lot simpler.

Also it should be easy to use:

```
from exreporter.credentials import GithubCredentials
from exreporter.stores import GithubStore
from exreporter import ExReporter
gc = GithubCredentials(user="username", repo="repo-name", auth_token="personal-token")
gs = GithubStore(credentials=gc)
reporter = ExReporter(store=gs, labels=['Bug'])

reporter.report()
```

1.1 Features

- Creates issues in issue trackers
- Aggregate same kind of issues
- Handle multiple occurrence

1.2 Usage

Usage reference available at <https://exreporter.readthedocs.org/en/latest/usage.html>

1.3 Documentation

Documentation is available at <https://exreporter.readthedocs.org/>.

1.4 TODOS

1. Add support for Bitbucket

Installation

At the command line:

```
$ easy_install exreporter
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv exreporter  
$ pip install exreporter
```

Usage

To use Exreporter in any project:

```
from exreporter import ExREporter
from exreporter.credentials import GithubCredentials
from exreporter.stores import GithubStore

gc = GithubCredentials(
    user="username", repo="reponame", auth_token="personaltoken")
gs = GithubStore(credentials=gc)
reporter = ExREporter(store=gs)

reporter.report()
```

3.1 Exreporter in Django Project

Exreporter requires following Django settings in `settings.py`:

```
EXREPORTER_GITHUB_USER = "username"
EXREPORTER_GITHUB_REPO = "reponame"
EXREPORTER_GITHUB_AUTH_TOKEN = "personaltoken"
EXREPORTER_GITHUB_LABELS = ['Bug']
```

And then add Exreporter's middleware in `settings.py`:

```
MIDDLEWARE_CLASSES = (
    ...
    'exreporter.contrib.django_middleware.ExreporterGithubMiddleware',
)
```

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/vedarthk/exreporter/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

Create issues in Github for internal server errors or exceptions in background tasks. could always use more documentation, whether as part of the official Create issues in Github for internal server errors or exceptions in background tasks. 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/vedarthk/exreporter/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 *exreporter* for local development.

1. Fork the *exreporter* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/exreporter.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 exreporter
$ cd exreporter/
$ 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 exreporter 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.

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, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/vedarthk/exreporter/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_exreporter
```

Credits

5.1 Development Lead

- Vedarth Kulkarni <vedarthk@vedarthz.in>

5.2 Contributors

None yet. Why not be the first?

History

0.1.0 (2014-12-11)

- First release on PyPI.

0.1.1 (2014-12-15)

- Change development status to Alpha.
- Also add contrib module - `django_middlewares`

0.1.2 (2015-04-04)

- Change public API
- Refactor github issue search

Indices and tables

- *genindex*
- *modindex*
- *search*