Gerritlib Documentation

Release 0.4.1.dev14

Openstack Infra

August 02, 2015

Contents

CHAPTER 1

README

Gerritlib is a Python library for interacting with Gerrit. It aims to provide a more conventionally pythonic way of managing a Gerrit instance.

To install:

```
$ sudo python setup.py install
```

Online documentation:

- http://gerritlib.readthedocs.org/en/latest/

1.1 Developers

Bug report:

- https://storyboard.openstack.org/#!/project/718

Repository:

- https://git.openstack.org/cgit/openstack-infra/gerritlib

Cloning:

```
git clone https://git.openstack.org/openstack-infra/gerritlib
```

Patches are submitted via Gerrit at:

- https://review.openstack.org/

Please do not submit GitHub pull requests, they will be automatically closed.

More details on how you can contribute is available on our wiki at:

- http://docs.openstack.org/infra/manual/developers.html

1.2 Writing a patch

We ask that all code submissions be pep8 and pyflakes clean. The easiest way to do that is to run tox before submitting code for review in Gerrit. It will run `pep8` and `pyflakes` in the same manner as the automated test suite that will run on proposed patchsets.

1.3 Installing without setup.py

Then install the required python packages using pip:

```
$ sudo pip install gerritlib
```

CHAPTER 2

Contents

2.1 API reference

class `gerritlib.gerrit.`**`GerritWatcher`**(*gerrit*, *username=None*, *hostname=None*, *port=None*, *keyfile=None*, *connection_attempts=-1*, *retry_delay=5*)

Create a GerritWatcher.

Parameters **`gerrit`** – A Gerrit instance to pass events to.

All other parameters are optional and if not supplied are sourced from the gerrit instance.

`log` **= <logging.Logger object>**

`run`()

class `gerritlib.gerrit.`**`Gerrit`**(*hostname*, *username*, *port=29418*, *keyfile=None*)

`log` **= <logging.Logger object>**

`startWatching`(*connection_attempts=-1*, *retry_delay=5*)

`addEvent`(*data*)

`getEvent`()

`createGroup`(*group*, *visible_to_all=True*, *owner=None*)

`createProject`(*project*, *require_change_id=True*, *empty_repo=False*, *description=None*)

`updateProject`(*project*, *update_key*, *update_value*)

`listProjects`(*show_description=False*)

`listGroups`(*verbose=False*)

`listPlugins`()

`getPlugins`()

`getVersion`()

`replicate`(*project='–all'*)

`review`(*project*, *change*, *message*, *action={}*)

`query`(*change*, *commit_msg=False*, *comments=False*)

`bulk_query`(*query*)

2.2 Contributing

If you would like to contribute to the development of OpenStack, you must follow the steps in this page:

> http://docs.openstack.org/infra/manual/developers.html

If you already have a good understanding of how the system works and your OpenStack accounts are set up, you can skip to the development workflow section of this documentation to learn how changes to OpenStack should be submitted for review via the Gerrit tool:

> http://docs.openstack.org/infra/manual/developers.html#development-workflow

Pull requests submitted through GitHub will be ignored.

Bugs should be filed on StoryBoard, not GitHub:

> https://storyboard.openstack.org/#!/project/718

To browse the latest code:

> https://git.openstack.org/cgit/openstack-infra/gerritlib/tree/

To clone the latest code:

> git clone git://git.openstack.org/openstack-infra/gerritlib

Code reviews are handled by gerrit: http://review.openstack.org

Use *git review* to submit patches (after creating a gerrit account that links to your launchpad account). Example:

```
# Do your commits
$ git review
# Enter your username if prompted
```

2.3 Installing

The module is known to pip and Debian-based distributions as `gerritlib`.

`pip`:

```
pip install gerritlib
```

`easy_install`:

```
easy_install gerritlib
```

The module has been packaged since Ubuntu Oneiric (11.10):

```
apt-get install gerritlib
```

And on Fedora 19 and later:

```
yum install gerritlib
```

For development:

```
python setup.py develop
```

2.3.1 Documentation

Documentation is included in the doc folder. To generate docs locally execute the command:

```
tox -e docs
```

The generated documentation is then available under doc/build/html/index.html.

2.3.2 Unit Tests

Unit tests are in the tests folder. To run the unit tests, execute the command:

```
tox -e py27
```

- Note: View tox.ini to run tests on other versions of Python.

2.4 Usage

Example usage:

```
import gerritlib.gerrit as gerrit
g = gerrit.Gerrit('gerrit_host', 'username', keyfile='/home/username/.ssh/id_rsa.pub')

# manage projects
g.createProject('test', description='a test project')
projects = g.listProjects()
print(projects)

# manage groups
g.createGroup('testers')
groups = g.listGroups()
print(groups)
```

Look at the API reference for more details.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search