
Zeus Python Client Documentation

Release 0.2.1

Cisco Systems, Inc.

June 06, 2015

1	Zeus Python Client	3
1.1	Features	3
2	Installation	5
3	Usage	7
3.1	Create a Zeus client	7
3.2	Logs	7
3.3	Metrics	8
3.4	Dates	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	12
4.4	Tips	13
5	Credits	15
5.1	Development Lead	15
5.2	Contributors	15
6	History	17
7	0.2.1 (2015-06-06)	19
8	0.2.0 (2015-06-04)	21
9	0.1.0 (2015-04-30)	23
10	Indices and tables	25

Contents:

Zeus Python Client

Python client for CiscoZeus.io. It allows a user to send and receive data to and from Zeus.

- Free software: Apache License
- Documentation: <https://cisco-zeus.readthedocs.org>.

1.1 Features

- Send logs and metrics to Zeus
- Query both logs and metrics

Installation

At the command line:

```
$ pip install cisco-zeus
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv zeus  
$ pip install cisco-zeus
```

Usage

3.1 Create a Zeus client

To use Zeus Python Client in a project:

```
from zeus import client
```

Create a ZeusClient object:

```
z = client.ZeusClient(USER_TOKEN, 'api.ciscozeus.io')
```

Now you are ready to start sending and querying logs and metrics. :D

3.2 Logs

3.2.1 Log name format

All log names must have only letter and numbers (A-Za-z0-9).

3.2.2 Send logs

You can send any key/value pair as a log. To send logs:

```
logs = [
    {"message": "My Test Log"},
    {"message": "My Second Test Log"}
]
z.sendLog("<LOG_NAME>", logs)
```

3.2.3 Query logs

To query logs:

```
z.getLog('<LOG_NAME>',
        pattern='*',
        from_date=123456789,
        to_date=126235344235,
```

```
offset=23,  
limit=10)
```

3.3 Metrics

3.3.1 Metric name format

All metric names must start with a letter or number (A-Za-z0-9) and can contain '.', '-', and '_'.

3.3.2 Send Metrics

You can send metrics as an array of timestamps and value dictionaries. To send metrics:

```
metrics = [  
    {"timestamp": 123541423,  
     "point": {"column1": 1, "column2": 2}},  
    {"timestamp": 123541424,  
     "point": {"column1": 3, "column2": 4}},  
    {"timestamp": 123541425,  
     "point": {"column1": 5, "column2": 6}}  
]  
z.sendMetric("<METRIC_NAME>", metrics)
```

If you omit the “timestamp” field, then current system time will be used as timestamp.

3.3.3 Query metric names

To query metric names:

```
z.getMetricNames(metric_name="<METRIC_NAME>", limit=10)
```

3.3.4 Query metrics

To query metrics:

```
z.getMetric(metric_name='ZeusTest',  
            aggregator='sum',  
            from_date=123456789,  
            to_date=126235344235,  
            group_interval='1m',  
            filter_condition='value > 90',  
            limit=10)
```

3.3.5 Delete metrics

To delete a metric:

```
z.deleteMetric('ZeusTest')
```

This will delete the entire metric from Zeus.

3.4 Dates

All dates must be in the Unix timestamp format.

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/CiscoZeus/python-zeusclient/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

Zeus Python Client could always use more documentation, whether as part of the official Zeus Python Client 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/CiscoZeus/python-zeusclient/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 *zeus* for local development.

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

```
$ git clone git@github.com:your_name_here/python-zeusclient.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 zeus
$ cd python-zeusclient/
$ 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:

```
$ tox
```

To get tox, just 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.

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 and 2.7 and for the Docs. Check https://github.com/CiscoZeus/python-zeusclient/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_zeus_client
```

Credits

5.1 Development Lead

- Marc Solanas Tarre <msolanas@cisco.com>

5.2 Contributors

- Johnu George <johnugeo@cisco.com>
- Kai Zhang <kazhang2@cisco.com>

History

0.2.1 (2015-06-06)

- Fixed some bugs. Ready for the Hackathon.

0.2.0 (2015-06-04)

- Added support for the new API, which includes metric deletion.

0.1.0 (2015-04-30)

- First release on PyPI.

Indices and tables

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