pygerduty Documentation

Release latest

September 02, 2015

Contents

1 Installation 3

2 Requirements 5

3 Documentation 7

4 Examples 9

Python Library for PagerDuty's v1 REST API.

This library is currently evolving and backwards compatibility cannot always be guaranteed at this time.

CHAPTER 1

Installation

You can install with `pip install pygerduty`.

If you want to install from source, then `python setup.py install`.

CHAPTER 2

Requirements

Pygerduty is tested against >= Python 2.5

`simplejson` is required on Python2.5

CHAPTER 3

Documentation

Pygerduty is a thin wrapper around PagerDuty's APIs. You will need to refer to the the PagerDuty Documentation for all available parameters to pass and all available attributes on responses.

The main methods available to resources are list, show, create, update, and delete. Not all resources have endpoints for all of the above methods. Again, refer to the PagerDuty Documentation to see all available endpoints.

Top level resources will be accessible via the PagerDuty object and nested resources available on containers returned from their parent resource.

CHAPTER 4

Examples

Instantiating a client:

```
import pygerduty
pager = pygerduty.PagerDuty("foobar", "SOMEAPIKEY123456")
```

Listing a resource:

```
for schedule in pager.schedules.list():
    print schedule.id, schedule.name

# PX7F8S3 Primary
# PJ48C0S Tertiary
# PCJ94SK Secondary
```

Getting a resource by ID:

```
schedule = pager.schedules.show("PX7F8S3")
```

Creating a resource:

```
user = next(pager.users.list(query="gary", limit=1))
override = schedule.overrides.create(
    start="2012-12-16", end="2012-12-17", user_id=user.id)
```

Delete a resource:

```
schedule.overrides.delete(override.id)
```

Updating a resource:

```
pagerduty.users.update(user.id, name="Gary Example")
```

Acknowledging a group by incidents:

```
me = next(pager.users.list(query="me@you.com", limit=1))
for incident in pagerduty.incidents.list(status='triggered'):
    incident.acknowledge(requester_id=me.id)
```