
circleclient Documentation

Release 0.1.6

Jakub Jarosz

April 09, 2015

1	Main Features	3
2	Contents:	5
2.1	Installation	5
2.2	Usage	5
2.3	Contributing	8
2.4	Credits	8
2.5	History	9
3	Feedback	11

Python client library for CircleCI API.

`circleclient` is a Python client library used for accessing CircleCI API to retrieve information about builds. It can also be used to start and stop builds. Full list of implemented features can be found in the usage section.

Main Features

- Start build
- Cancel build
- Retry build
- Run parametrized builds
- Retrieve information about user
- Retrieve information about build(s)
- List followed repositories
- List build artifacts
- Clear build cache

Contents:

2.1 Installation

At the command line either via pip:

```
$ pip install circleclient
```

If you have virtualenvwrapper installed:

```
$ mkvirtualenv circleclient
$ pip install circleclient
```

Or, if you have virtualenv installed:

```
$ virtualenv circleclient-venv
$ source circleclient-venv/bin/activate
$ pip install circleclient
```

2.2 Usage

2.2.1 Retrieve information about User

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Retrieve User data
client.user.info()
```

2.2.2 List projects followed by the user

```
import os
from circleclient import circleclient
```

```
token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Retrieve information about projects
client.projects.list_projects()
```

2.2.3 Trigger new build

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Trigger build
client.build.trigger('<username>', '<project_name>', '<branch>')
```

2.2.4 Trigger new parametrized build

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Trigger parametrized build
client.build.trigger('<username>', '<project_name>', '<branch>', '<PARAM1>'='<VAL1>')
```

2.2.5 Cancel running build

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Cancel build
client.build.cancel('<username>', '<project_name>', '<build_number>')
```

2.2.6 Retry build

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Retry build
client.build.retry('<username>', '<project_name>', '<build_number>')
```

2.2.7 List build artifacts

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# List build artifacts
client.build.artifacts('<username>', '<project_name>', '<build_number>')
```

2.2.8 Retrieve build status

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Retrieve build status
client.build.status('<username>', '<project_name>', '<build_number>')
```

2.2.9 Retrieve information about builds across all projects

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Retrieve build status
# Default limit=30, offset=0
client.build.recent_all_projects(limit=<int>, offset=0)
client.build.recent_all_projects()
```

2.2.10 Retrieve information about recent build(s)

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(token)

# Retrieve build status
# Default limit=30, offset=0, branch=None
client.build.recent('<username>', '<project>', limit='<int>', offset='<int>')

# Retrieve last 10 builds of branch master
client.build.recent('<username>', '<project>', limit=10, branch='master')

# Retrieve last build of branch develop
client.build.recent('<username>', '<project>', branch='develop')
```

2.2.11 Clear build cache

```
import os
from circleclient import circleclient

token = os.environ['API_TOKEN']
client = circleclient.CircleClient(api_token=token)

# Clear build cache
client.cache.clear(username='<username>', project='<project_name>')
```

2.3 Contributing

2.3.1 Contributing Code

- A good patch:
 - is clear
 - works across all supported versions of Python
 - follows the existing style of the code base (PEP-8)
 - has docstrings (we use Goole style docstrings)
 - has comments included as needed
- A test case that demonstrates the previous flaw that now passes with the included patch
- If it adds/changes a public API, it must also include documentation for those changes

2.3.2 Reporting An Issue/Feature

- Check to see if there's an existing issue/pull request for the bug/feature. All issues are at <https://github.com/qba73/circleclient/issues> and pull reqs are at <https://github.com/qba73/circleclient/pulls>
- If there isn't an existing issue there, please file an issue. The ideal report includes:
 - A description of the problem/suggestion
 - Steps to recreate the bug/issue
 - Versions of your Python interpreter

2.4 Credits

Development Lead

- Jakub Jarosz (@qba73)

Contributors

- Jason Boyles (@JasonBoyles)
- lucy (@diN0bot)

2.5 History

2.5.1 0.1.6 (2015-09-04)

- Updated documentation

2.5.2 0.1.5 (2015-31-03)

- Filter recent builds by build status: completed, successful, failed, running

2.5.3 0.1.4 (2014-21-09)

- Retrieving information about single and multiple builds.
- Retrieving a limited number of builds with optional offset.
- Retrieving build information about specific branch.

2.5.4 0.1.3 (2014-25-07)

- Add support for parametrized builds
- Add support for listing build artifacts

2.5.5 0.1.2 (2014-07-07)

- Add support for clearing cache

2.5.6 0.1.1 (2014-30-06)

- Add support for retrying builds

2.5.7 0.1.0 (2014-29-06)

- First release on PyPI.

Feedback

If you have any suggestions or questions about **circleclient** feel free to email me at qba73@postpro.net.

If you encounter any errors or problems with **circleclient**, please let me know! Open an Issue at the GitHub <http://github.com/qba73/circleclient> main repository.