python-crowd Documentation

Release 0.7

Alexander Else

September 11, 2016

Contents

python-crowd is a python client library to the Atlassian Crowd REST API.

This library may be useful to you if you wish to create an application that authenticates users against Crowd or integrate with Crowd's SSO.

This project has no official status within Atlassian and is provided as a service to the programming community. I hope that you find it useful.

CHAPTER 1

Documentation

Docs are built automatically by sphinx. You can build them yourself in the doc directory or access them at <http://aelse.github.com/python-crowd/>.

CHAPTER 2

Examples

Some example invocations are provided in the examples directory, and are included in the built documentation.

The unit tests may also be instructive.

CHAPTER 3

License

BSD 2-Clause License

Copyright (c) 2012-2013, Alexander Else All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents:

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

CHAPTER 5

Simple Authentication Test

```python

import crowd
import os, sys, getpass

app_url = 'http://my.crowd.server:8095/crowd/'
app_user = 'testapp'
app_pass = 'testpass'

# Create the reusable Crowd object
cs = crowd.CrowdServer(app_url, app_user, app_pass)

if len(sys.argv) > 1:
    username = sys.argv[1]
else:
    username = os.environ['USER']

password = getpass.getpass(prompt='Enter password for %s: ' % username)

success = cs.auth_user(username, password)
if success:
    print 'Successfully authenticated.'
else:
    print 'Failed to authenticate.'
```

CHAPTER 6

Simple Token Authentication

```python
import crowd
import os, sys, getpass

app_url = 'http://my.crowd.server:8095/crowd/'
app_user = 'testapp'
app_pass = 'testpass'

# Create the reusable Crowd object
cs = crowd.CrowdServer(app_url, app_user, app_pass)

if len(sys.argv) > 1:
    username = sys.argv[1]
else:
    username = os.environ['USER']

password = getpass.getpass(prompt='Enter password for %s: ' % username)

# Create a session. The important bit is the token.
session = cs.get_session(username, password)
if session:
    print 'Created a session, token %s' % session['token']
else:
    print 'Failed to authenticate.'
    sys.exit(1)

# Check that the token is valid (and of course it should be).
success = cs.validate_session(session['token'])
if success:
    print 'Authenticated session token.'
else:
    print 'Failed to authenticate token.'
```