pyucwa Documentation

Release 0.3.0

Anthony Shaw

Jun 30, 2017

Contents

Contents:

CHAPTER 1

pyucwa

Skype for Business UCWA API client

- Free software: Apache 2 license
- Documentation: https://ucwa.readthedocs.org.

Usage

Setup your tenant

Follow the steps in https://msdn.microsoft.com/en-us/office/office365/howto/add-common-consent-manually

Enter the pool for your tenant by visiting the URL : https://webdir.online.lync.com/Autodiscover/AutodiscoverService.svc/root?originalDomain=parliamentfunksterhotmail.onmicrosoft.com with your domain.

Create a file config.yml with similar details

```
redirect_uri: "http://127.0.0.1:5000"
client_id: "0b78a9be-6b65-1234-b8e6-a0b21a8672c3"
secret: "jPpYkK+sdf3423r="
domain: "mydomain.onmicrosoft.com"
app_id: "https://mydomain.onmicrosoft.com/bot"
```

Start the web server

    python -m ucwa.http

Run a login session to get a token for the application

    python authhelper.py

This will open the browser, get you to login to Office 365 and then create an instance session with a UCWA server in O365/Skype for Business online

You can then run app.py to stream events

    python app.py

Extend app.py to do what you want to the events, like have a chat with other people or integrate into your bot framework.

Features

- TODO

Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER 2

Installation

At the command line:

```
$ easy_install ucwa
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv ucwa
$ pip install ucwa
```

Configuring SfB Online

Follow the steps in https://msdn.microsoft.com/en-us/office/office365/howto/add-common-consent-manually

Enter the pool for your tenant by visiting the URL : https://webdir.online.lync.com/Autodiscover/AutodiscoverService.svc/root?originalDomain=parliamentfunksterhotmail.onmicrosoft.com with your domain.

You will need to know for the given user which Skype pool they are running on, login to the Office 365 Admin Center, go to the Lync/SfB Admin Console.

CHAPTER 3

Usage

To use pyucwa in a project:

```
import ucwa
```

CHAPTER 4

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:

Types of Contributions

Report Bugs

Report bugs at https://github.com/tonybaloney/pyucwa/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.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

Write Documentation

pyucwa could always use more documentation, whether as part of the official pyucwa docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/tonybaloney/pyucwa/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 :)

Get Started!

Ready to contribute? Here's how to set up *ucwa* for local development.

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

```
$ git clone git@github.com:your_name_here/pyucwa.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 pyucwa
$ cd pyucwa/
$ 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:

```
$ flake8 ucwa tests
$ python setup.py test
$ tox
```

   To get flake8 and tox, just pip install them 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.

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, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/tonybaloney/pyucwa/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_ucwa
```

CHAPTER 5

Credits

Development Lead

- Anthony Shaw <anthonyshaw@apache.org>

Contributors

None yet. Why not be the first?

CHAPTER 6

History

0.1.0 (2016-04-04)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search