yowsup celery Documentation

Release 0.2.1

Juan Madurga

March 12, 2016

Contents

Contents:

CHAPTER 1

yowsup celery

CI: PyPI: Docs: Yowsup integrated in a celery architecture

- Free software: ISC license
- Documentation: https://yowsup-celery.readthedocs.org.

1.1 Features

- **Celery app adapted to Yowsup**
  - Bootstep added to worker to initialize Yowsup and stopping when TERM (sometimes kill -9 is necessary)
  - Options added to execute workers with different Whatsapp accounts
  - Only works with gevent and threads as yowsup socket is shared between tasks
- **Yowsup features included:**
  - Connect/Disconnect
  - Send Text, Image and Audio Messages
  - Receive Messages, Acks and Receipts

CHAPTER 2

Installation

At the command line:

```
    $ pip install -e git+https://github.com/jlmadurga/yowsup.git@issue_1181#egg=yowsup
$ pip install yowsup-celery
```

At the moment only works with yowsup fork https://github.com/jlmadurga/yowsup.git@issue_1181

CHAPTER 3

Usage

Create a Yowcelery app:

```
import yowsup_celery

app = YowCelery('example',
     broker='amqp://',
     #backend='amqp://',
     include=['yowsup_celery.tasks'])
```

You can add other layer between core layers and celery top layer:

```
   app.conf.update(
      TOP_LAYERS=('yowsup_ext.layers.store.layer.YowStorageLayer',)
)
```

Then to launch worker:

```
$ celery -A proj worker -P gevent -c 2 -l info --yowconfig conf_wasap
```

Yowsup celery worker only works with gevent and threads pools. Yowsup asyncore socket need to be shared between tasks.

You can see the new worker options: celery -A proj worker –help:

```
--yowconfig=CONFIG    Path to config file containing authentication info.
--yowlogin=phone:b64password
                      WhatsApp login credentials, in the format
                      phonenumber:password,                      where
                      password is base64 encoded.
--yowunmoxie          Disable E2E Encryption
```

Just call tasks as other celery app:

```
from yowsup_celery import tasks

tasks.connect.delay()
tasks.send_message.delay("341234567", "New message sent")
taks.disconnect.delay()
```

You can have a multiple workers for different phone numbers routing each worker to its queue:

```
$ celery -A proj worker -P gevent -c 2 -l info --yowconfig conf_wasap_number1 -Q number1
```

When calling tasks queue to the queue desired:

```
taks.connect.apply_async(queue="number1")
```

If you want to use celery as daemon just add yowconfig path in configuration:

```
   app.conf.update(
      YOWSUPCONFIG='path/to/yowsupconfig/file'
)
```

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:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/jlmadurga/yowsup-celery/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

yowsup celery could always use more documentation, whether as part of the official yowsup celery 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/jlmadurga/yowsup-celery/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 *yowsup-celery* for local development.

1. Fork the *yowsup-celery* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/yowsup-celery.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 yowsup-celery
$ cd yowsup-celery/
$ 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 yowsup_celery tests
$ python setup.py test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv. You can use Makefile instead:

```
$ make lint
$ make test
$ make test-all
```

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, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/jlmadurga/yowsup-celery/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_yowsup-celery
```

CHAPTER 5

Credits

5.1 Development Lead

- Juan Madurga <jlmadurga@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

CHAPTER 7

0.1.0 (2015-31-12)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search