Flask Hypertable Documentation

Release 0.3.0

Fairiz Azizi

September 15, 2015

Contents

Contents:

CHAPTER 1

Installation

At the command line:

```
$ easy_install flask_hypertable
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv flask_hypertable
$ pip install flask_hypertable
```

CHAPTER 2

Usage

This package provides two extension points

- `FlaskHypertable`

  Provides a basic `ThriftClient` connection to `Hypertable`. The connection is opened and closed for every HTTP request.

- `FlaskPooledHypertable`

  Extends the base `FlaskHypertable` class to offer transparent connection pooling. The connection is opened only once for the lifetime of this instance. The `ThriftClient` is reused across HTTP requests.

2.1 Configuration

Your configuration should be declared within your Flask config:

```
HYPERTABLE_HOST = "localhost"
HYPERTABLE_PORT = 38080
HYPERTABLE_TIMEOUT_MSECS = 5000

################
#if using FlaskPooledHypertable

#the size of the pool to maintain, you probably want at least 1
HYPERTABLE_POOL_SIZE = 5

#if all connections are busy, create this many extra connections
#0 disables
HYPERTABLE_MAX_OVERFLOW = 10
```

2.2 Flask App Extension

To create the `FlaskHypertable` instance within your application:

```
from flask import Flask
from flask_hypertable import FlaskHypertable

app = Flask(__name__)
ht = FlaskHypertable(app)
```

or:

```
from flask import Flask
from flask_hypertable import FlaskHypertable

ht = FlaskHypertable()

def create_app():
    app = Flask(__name__)
    ht.init_app(app)
    return app
```

Or if you would like to use pooled connections:

```
from flask import Flask
from flask_hypertable import FlaskPooledHypertable

app = Flask(__name__)
ht = FlaskPooledHypertable(app)
```

or:

```
from flask import Flask
from flask_hypertable import FlaskPooledHypertable

ht = FlaskPooledHypertable()

def create_app():
    app = Flask(__name__)
    ht.init_app(app)
    return app
```

The Flask extension provides the `flask_hypertable.ManagedThriftClient` instance as the `ht.connection` member attribute.

To properly shutdown the `FlaskPooledHypertable` the *close_app* method is automatically called at exit, or, you may call it manually.

2.3 Basic Usage

The following demonstrates using the extension at its most basic level.

See http://hypertable.com/documentation/code_examples/python for more information.

```
#...

client = ht.connection

#...
#in some method
ns = client.namespace_open("test")
res = client.hql_query(ns, "select * from foo")
for cell in res.cells: print cell
client.close_namespace(ns)

#...
#in some other method (during the same request)
ns = client.namespace_open("test")
```

```
res = client.hql_query(ns, "select * from bar")
for cell in res.cells: print cell
client.close_namespace(ns)
```

2.4 With Usage

The client also supports `with` semantics

```
#...

with ht as client:
    ns = client.namespace_open("test")
    client.hql_query(ns, "select * from foo")
```

2.5 Managed Namespaces

The above example suffers by having duplicate boiler plate code surrounding opening the namespace.

It also suffers from the fact that each method will end up opening and closing namespaces more than once within a request.

To alleviate this, the `FlaskHypertable.connection` can help you manage your namespaces. This is available through a helper member attribute called `mns`.

This helper provides a method to open or reuse previously created namespaces.

In this manner, we also prevent unnecessary roundtrips to Hypertable.

The above would shorten to something like this:

```
#...

client = ht.connection

#in some method
res = client.hql_query(client.mns['test'], "select * from foo")

#in some other method (during the same request)
res = client.hql_query(client.mns['test'], "select * from bar")
```

In the above example, `client.ns['test']` is a shortcut to `client.mns.open_namespace('test')`.

To close the namespace:

```
client.mns.close_namespace('test')

#or

client.close()
```

2.6 Troubleshooting

- ThriftClient.open and close seems to be calling too much

- Use the `FlaskPooledHypertable` instead (new since v0.2.0)
- Try changing the pool configuration settings.

Did you remember to call the `FlaskHypertable.init_app(app)` method when setting up your Flask App? If not, the extension will fall back to creating itself in each context. See http://flask.pocoo.org/docs/extensiondev/ for more information.

CHAPTER 3

API

3.1 Flask Hypertable

A Flask extension for Hypertable over Thrift.

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/coderfi/flask_hypertable/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

Flask Hypertable could always use more documentation, whether as part of the official Flask Hypertable 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/coderfi/flask_hypertable/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 *flask_hypertable* for local development.

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

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

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 and for PyPy. Check https://travis-ci.org/coderfi/flask_hypertable/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_flask_hypertable
```

CHAPTER 5

Credits

5.1 Development Lead

- Fairiz Azizi <coderfi@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

Roadmap

6.1 dev

- include libHyperPython somehow?
- extend ManagedThriftClient with mutator creation helpers
- ORM? Draw on MongoKit, MongoEngine, SQLAlchemy as inspiration
- tests

CHAPTER 7

History

7.1 dev

Released: Ongoing

7.1.1 docs

- **[docs]** Updated CHANGES. ¶

7.2 0.3.0

Released: 2014-03-30

- **[project]** Added HYPERTABLE_TIMEOUT_MSECS option (defaults to 5000 msecs) Removed superflous _qo member from FlaskPooledHypertable FlaskPooledHypertable constructor now takes an optional `qClass` override. Now supports `with` semantics. ¶

7.3 0.2.0

Released: 2014-03-26

- **[project]** Now supports connection pooling with FlaskPooledHypertable! Added the 'mns' helper object in order to support managed ¶

7.4 0.1.4

Released: 2014-03-23

- **[project]** Added Hypertable 0.9.5.6 version info to README.rst ¶

7.5 0.1.3

Released: 2014-03-22

- **[project]** Added Hypertable license information. ¶

7.6 0.1.2

Released: 2014-03-22

- **[project]** First commit to github

  Docs available on readthedocs.org

  CI available at travis-ci.org

  Added project to badge.fury.io

  Added project to coveralls.io

  Added project to oholoh.net ¶

- `FlaskHypertable` `hypertable.thrift.ThriftClient` Flask extension.

CHAPTER 8

Installation

```
pip install Flask-Hypertable
```

Or if you *must* use easy_install:

```
alias easy_install="pip install $1"
easy_install Flask-Hypertable
```

CHAPTER 9

Configuration

Your configuration should be declared within your Flask config.

```
HYPERTABLE_HOST = "localhost"
HYPERTABLE_PORT = 38080
```

To create the Hypertable instance within your application

```
from flask import Flask
from flask_hypertable import FlaskHypertable

app = Flask(__name__)
ht = FlaskHypertable(app)
```

or

```
from flask import Flask
from flask_hypertable import FlaskHypertable

ht = FlaskHypertable()

def create_app():
    app = Flask(__name__)
    ht.init_app(app)
    return app
```

Hypertable	0.9.5.6 (other versions likely to work) http://hypertable.com/documentation/reference_manual/thrift_api
Thrift	https://thrift.apache.org/docs/
Python support	Python 2.7
Source	https://github.com/coderfi/flask-hypertable
Docs	http://flask-hypertable.rtfd.org
Changelog	http://flask-hypertable.readthedocs.org/en/latest/history.html
API	http://flask-hypertable.readthedocs.org/en/latest/api.html
Issues	https://github.com/coderfi/Flask-Hypertable/issues
Travis	http://travis-ci.org/coderfi/Flask-Hypertable
Test coverage	https://coveralls.io/r/coderfi/Flask-Hypertable
pypi	https://pypi.python.org/pypi/Flask-Hypertable
Ohloh	https://www.ohloh.net/p/Flask-Hypertable
License	BSD.
git repo	`$ git clone https://github.com/coderfi/Flask-Hyper`

install dev	`$ git clone https://github.com/coderfi/Flask-Hyper`
	`$ cd ./flask-hypertable`
	`$ virtualenv .env`
	`$ source .env/bin/activate`
	`$ pip install -e .`

tests	`$ python setup.py test`
	or
	`$ tox`
	or
	`$ python run-tests.py`

CHAPTER 10

About This Project

Project started with cookiecutter-pypackage.

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index