
aioh2 Documentation

Release 0.2.0

Fantix King

March 08, 2016

1	aioh2	3
1.1	Features	3
1.2	Example	3
1.3	Credits	4
2	Installation	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.1.0 (2016-2-6)	15
7	Indices and tables	17

Contents:

HTTP/2 implementation with [hyper-h2](#) on Python 3 asyncio.

- Free software: BSD license
- Documentation: <https://aiohttp.readthedocs.org>.

1.1 Features

- Asynchronous HTTP/2 client and server
- Multiplexing streams of data with managed flow and priority control
- Optional tickless health check
- More to come

Non-features:

- Request/Response wrappers
- Web server, dispatcher, cookie, etc
- HTTP/2 upgrade

1.2 Example

A server saying hello:

```
# Server request handler
async def on_connected(proto):
    while True:
        # Receive a request from queue
        stream_id, headers = await proto.recv_request()

        # Send response headers
        await proto.start_response(stream_id, {'status': '200'})

        # Response body starts with "hello"
        await proto.send_data(stream_id, b'hello, ')

        # Read all request body as a name from client side
        name = await proto.read_stream(stream_id, -1)
```

```
# Amend response body with the name
await proto.send_data(stream_id, name)

# Send trailers with the length of the name
await proto.send_trailers(stream_id, {'len': str(len(name))})

# Start server on random port, with maximum concurrent requests of 3
server = await aioh2.start_server(
    lambda p: asyncio.get_event_loop().create_task(on_connected(p)),
    port=0, concurrency=3)
port = server.sockets[0].getsockname()[1]
```

And a client to try out:

```
# Open client connection
client = await aioh2.open_connection('0.0.0.0', port,
                                     functional_timeout=0.1)

# Optionally wait for an ack of tickless ping - a.k.a. until functional
await asyncio.sleep(0.1) # simulate client being busy with something else
rtt = await client.wait_functional()
if rtt:
    print('Round-trip time: %.1fms' % (rtt * 1000))

# Start request with headers
stream_id = await client.start_request(
    {'method': 'GET', 'path': '/index.html'})

# Send my name "world" as whole request body
await client.send_data(stream_id, b'world', end_stream=True)

# Receive response headers
headers = await client.recv_response(stream_id)
print('Response headers:', headers)

# Read all response body
resp = await client.read_stream(stream_id, -1)
print('Response body:', resp)

# Read response trailers
trailers = await client.recv_trailers(stream_id)
print('Response trailers:', trailers)
```

Above example can be found at *examples/core.py*.

1.3 Credits

A big thanks to the great library [hyper-h2](#) from [Cory Benfield](#).

[DecentFoX Studio](#) is a software outsourcing company delivering high-quality web-based products and mobile apps for global customers with agile methodology, focusing on bleeding-edge technologies and fast-developing scalable architectures.

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

Installation

At the command line:

```
$ easy_install aioh2
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv aioh2
$ pip install aioh2
```

Usage

To use aioh2 in a project:

```
import aioh2
```

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/decentfox/aioh2/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

aioh2 could always use more documentation, whether as part of the official aioh2 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/decentfox/aioh2/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 *aioh2* for local development.

1. Fork the *aioh2* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/aioh2.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 aioh2
$ cd aioh2/
$ 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 aioh2 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 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/decentfox/aioh2/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_aioh2
```

Credits

5.1 Development Lead

- Fantix King <fantix.king@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

6.1 0.1.0 (2016-2-6)

- First release on PyPI.

Indices and tables

- `genindex`
- `modindex`
- `search`