hassdevice

Release 0.1.4

Mar 03, 2019

Contents

CHAPTER 1

Overview

docs	
tests	
package	

A library for building MQTT devices for HomeAssistant

- Free software: Apache Software License 2.0

1.1 Installation

```
pip install hassdevice
```

1.2 Documentation

https://python-hassdevice.readthedocs.io/

1.3 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	`set PYTEST_ADDOPTS=--cov-append` `tox`
Other	`PYTEST_ADDOPTS=--cov-append tox`

CHAPTER 2

Installation

At the command line:

```
pip install hassdevice
```

CHAPTER 3

Usage

To use hassdevice in a project:

```
import hassdevice
```

CHAPTER 4

Reference

4.1 Devices

4.1.1 hassdevice.devices

class hassdevice.devices.**Switch**(*name*, *entity_id*)

An MQTT switch

__init__(*name*, *entity_id*)

Parameters

- **name** – The display name to use in HomeAssistant
- **entity_id** – The entity_id to use in HomeAssistant

base_topic

command_topic

config

config_topic

connect(*mqtt_client*, *discovery_prefix='homeassistant'*, *node_id=None*)

Connect this device to an MQTT broker

Parameters

- **mqtt_client** – A connected MQTT client
- **discovery_prefix** – The discovery prefix set in the HomeAssistant config
- **node_id** – Will be included in the MQTT topics if set

on_state_change(*new_state*)

Called when a state update request is recieved from the broker

Parameters **new_state** – The state to set

payload_off
Payload to use to indicate the switch is off. Defaults to `"OFF"`

payload_on
Payload to use to indicate the switch is on. Defaults to `"ON"`

retain
Should the messages sent to the broker have the 'retain' flag set. Defaults to `True`

state
The state of the switch. Must be one of `self.payload_on` or `self.payload_off`

Getter The last state the switch was set to (May be `None` if the state has never been set)

Setter Record the state change, and report it to the broker

state_topic

CHAPTER 5

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

5.1 Bug reports

When 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.

5.2 Documentation improvements

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

5.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/insertjokehere/python-hassdevice/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 code contributions are welcome :)

5.4 Development

To set up *python-hassdevice* for local development:

1. Fork python-hassdevice (look for the "Fork" button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/python-hassdevice.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

4. When you're done making changes, run all the checks, doc builder and spell checker with tox one command:

```
tox
```

5. 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
```

6. Submit a pull request through the GitHub website.

5.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)[1].
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

[1] If you don't have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though ...

CHAPTER 6

Authors

- Will Hughes - https://www.willhughes.name

CHAPTER 7

Changelog

7.1 0.0.2 (2017-07-17)

- Drop Python 2.7 support
- Add *hassdevice.hosts.SimpleMQTTHost*

7.2 0.0.1 (2017-07-10)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

h

Index

Symbols

B

C

H

O

P

R

S