
keenmqtt Documentation

Release

Author

August 15, 2015

1	keenmqtt package	3
1.1	Submodules	3
1.2	keenmqtt.app module	3
1.3	keenmqtt.keenmqtt module	3
1.4	Module contents	6
2	Indices and tables	7
	Python Module Index	9

Contents:

keenmqtt package

1.1 Submodules

1.2 keenmqtt.app module

Functions and classes for a command line app version of keenmqtt

1.3 keenmqtt.keenmqtt module

Keen mqtt relay class

exception `keenmqtt.keenmqtt.BackgroundRunningException`

Bases: `exceptions.Exception`

Used when the user tries to run in the foreground whilst a background loop is already running.

class `keenmqtt.keenmqtt.KeenMQTT`

add_collection_mapping (*sub*, *collection*)

Add a subscription to event collection mapping.

This will override existing subscriptions if present.

Parameters

- **sub** (*str*) – The string subscription pattern.
- **collection** (*str*) – The sting event collection.

Returns None

connect_keen (*settings*)

Setup the Keen IO client.

Parameters **Optional[dict]** (*settings*) – The settings object, such as one read from config.yaml

Returns None

connect_mqtt_client (*settings*)

Setup MQTT client.

Please note that the MQTT client will not actually connect until either `step` or `start` has been called.

Parameters **Optional[dict]** (*settings*) – The settings object, such as one read from config.yaml

Returns None

decode_payload (*topic, payload*)

Decode the payload of an incoming MQTT payload.

By default a JSON object is expected, however this method can be overridden to provide alternative means to extract a MQTT payload. For example, a binary format could be extracted here.

Parameters

- **topic** (*str*) – The topic string.
- **payload** (*str*) – Raw MQTT payload.

Returns An array of dictionaries containing the decoded MQTT payload.

Raises `ValueError` – Whent the JSON payload cannot be parsed.

get_time (*topic, message*)

Get the timestamp to send to Keen IO.

This method is used to extract the timestamp from the MQTT message if required, or to generate a timestamp. By default, the current time will be fetched.

Parameters

- **topic** (*str*) – The topic string.
- **message** (*dict*) – The message dictionary.

Returns A string containing ISO-8601 string.

Return type `str`

on_mqtt_connect (*c, client, userdata, rc*)

Called when an MQTT connection is made.

See the Paha MQTT client documentation `on_connect` documentation for arguments.

on_mqtt_message (*mosq, obj, mqtt_message*)

Called when an MQTT message is recieved.

See the Paha MQTT client documentation `on_message` documentation for arguments.

process_collection (*topic, message*)

Assign a collection to the MQTT message.

By default will find a matching topic in the `collection_mapping` dictionary and return the associated string. Could also be based on event contents.

Parameters

- **event** (*dict*) – The event dictionary for this mqtt message.
- **topic** (*str*) – The topic string.

Returns A string indicating the Keen IO collection which this event should be pushed to, or false if a matching event collection could not be found.

Return type `str`

process_payload (*event, topic, message*)

Process an incoming MQTT message's payload.

Perform any required translations to the payload of the MQTT message, such as removing

Parameters

- **event** (*dict*) – The event dictionary for this mqtt message.
- **topic** (*str*) – The topic string.
- **message** (*dict*) – the decoded MQTT payload

Returns A Boolean indicating if this message should continue through the pipeline. Return `False` to cancel the processing of this event and stop it from being saved in Keen IO.

Return type bool

process_time (*event, topic, message*)

Process the timestamp which will be sent to Keen IO.

If the MQTT message contains time information which should be used instead of the event being timestamped by Keen IO, set it here.

Parameters

- **event** (*dict*) – The event dictionary for this mqtt message.
- **topic** (*str*) – The topic string.
- **message** (*dict*) – The message dictionary.

Returns A Boolean indicating if this message should continue through the pipeline. Return `False` to cancel the processing of this event and stop it from being saved in Keen IO.

Return type bool

process_topic (*event, topic*)

Process an incoming MQTT message's topic string.

If the topic contains pertinent information, such as the device ID or location, this method can be overridden to perform any translation. By default, a key called `mqtt_topic` containing the topic string will be added to the event dictionary.

Parameters

- **event** (*dict*) – The event dictionary for this mqtt message.
- **topic** (*str*) – The topic string.

Returns A Boolean indicating if this message should continue through the pipeline. Return `False` to cancel the processing of this event and stop it from being saved in keen.

Return type bool

push_event (*collection, event*)

Thin wrapper around Keen IO API object.

Parameters

- **collection** (*str*) – The collection string to push to
- **event** (*dict*) – The complete event to push

Returns None

register_subscriptions ()

This should always be called since re-subscribes after any unexpected disconnects.

setup (*mqtt_client=None, keen_client=None, settings=None*)

Setup the clients for this instance.

Normally called with a settings object containing *keen* and *mqtt* keys with dictionary values of settings.

Parameters

- **Optional[class]** (*keen_client*) – An instance of an Paho MQTT client class.
- **Optional[class]** – An instance of a KeenClient.
- **Optional[dict]** (*settings*) – A settings dict, normally loaded from a config.yaml file.

Returns None

start ()

Automatically loop in a background thread.

step ()

Do a single MQTT step.

Use this if you're not running keenmqtt in a background thread with `start/stop`

stop ()

Disconnect and stop.

1.4 Module contents

Indices and tables

- *genindex*
- *modindex*
- *search*

k

keenmqtt, 6

keenmqtt.app, 3

keenmqtt.keenmqtt, 3

A

add_collection_mapping() (keenmqtt.keenmqtt.KeenMQTT method), 3

B

BackgroundRunningException, 3

C

connect_keen() (keenmqtt.keenmqtt.KeenMQTT method), 3

connect_mqtt_client() (keenmqtt.keenmqtt.KeenMQTT method), 3

D

decode_payload() (keenmqtt.keenmqtt.KeenMQTT method), 4

G

get_time() (keenmqtt.keenmqtt.KeenMQTT method), 4

K

KeenMQTT (class in keenmqtt.keenmqtt), 3

keenmqtt (module), 6

keenmqtt.app (module), 3

keenmqtt.keenmqtt (module), 3

O

on_mqtt_connect() (keenmqtt.keenmqtt.KeenMQTT method), 4

on_mqtt_message() (keenmqtt.keenmqtt.KeenMQTT method), 4

P

process_collection() (keenmqtt.keenmqtt.KeenMQTT method), 4

process_payload() (keenmqtt.keenmqtt.KeenMQTT method), 4

process_time() (keenmqtt.keenmqtt.KeenMQTT method), 5

process_topic() (keenmqtt.keenmqtt.KeenMQTT method), 5

push_event() (keenmqtt.keenmqtt.KeenMQTT method), 5

R

register_subscriptions() (keenmqtt.keenmqtt.KeenMQTT method), 5

S

setup() (keenmqtt.keenmqtt.KeenMQTT method), 5

start() (keenmqtt.keenmqtt.KeenMQTT method), 6

step() (keenmqtt.keenmqtt.KeenMQTT method), 6

stop() (keenmqtt.keenmqtt.KeenMQTT method), 6