
sqspy
Release 1.0.0

hjpotter92

Mar 27, 2021

CONTENTS:

- 1 Introduction** **1**
- 2 Install** **3**
- 3 Usage** **5**
- 4 Why** **7**
 - 4.1 Changelog 7
 - 4.2 sqspy package 8
- 5 Indices and tables** **13**
- Python Module Index** **15**
- Index** **17**

INTRODUCTION

A more pythonic approach to SQS producer/consumer utilities. Heavily inspired from the [the pySqsListener](#) package.

CHAPTER
TWO

INSTALL

```
pip install sqspy
```


USAGE

```
from sqspy import Consumer

class MyWorker(Consumer):
    def handle_message(self, body, attributes, message_attributes):
        print(body)

listener = MyWorker('Q1', error_queue='EQ1')
listener.listen()
```

More documentation coming soon.

WHY

The mentioned project had a few issues which I faced while trying to implement at my organisation. The local environment testing setup was very flaky. The signatures for `sqs_listener` and `sqs_producer` were very different from each other.

This rewrite supports **python 3.6+ versions only**, and makes use of a lot of newer python features. It also makes use of service resources (for lazy calls) from the boto3 library instead of making calls via the low level client.

4.1 Changelog

4.1.1 1.0.0 (2021-03-27)

- Add self-documentation using sphinx.
- Setup pipeline for readthedocs.
- Update contact email address.
- Update related links for pypi release page.
- Integrate with PyUp.

4.1.2 0.1.0 (2020-10-28)

- Setup tests using Travis CI.
- Add test coverage reports.

4.1.3 0.0.1 (2020-07-22)

- Initial release.
- Barebone working model of `Consumer` and `Producer`.

4.2 sqspy package

4.2.1 Submodules

4.2.2 sqspy._base module

class sqspy._base.Base (**kwargs)

Bases: object

Base class initialisation to setup aws credentials.

To make use of instance roles when deploying to AWS infrastructure, leave the `aws_*` keys blank (None).

Parameters

- **aws_access_key_id** (*str*) – AWS access key credential.
- **aws_secret_access_key** (*str*) – AWS access key credential.
- **profile_name** (*str*) – Local AWS credential profile name.
- **region_name** (*str*) – AWS region for resources.
- **endpoint_url** (*str*) – Custom endpoint URL for usage.

QUEUE_VISIBILITY_TIMEOUT: *str* = '600'

Message's visibility timeout in seconds. See [Visibility Timeout](#) in *Amazon Simple Queue Service Developer Guide* for more information.

create_queue (*name: str, attributes: Dict[str, Any]*)

Create a Queue resource.

For more information, check `SQS.ServiceResource.create_queue()`

Parameters

- **name** (*str*) – Sent as the *QueueName* to the boto3 method.
- **attributes** (*dict[str, str]*) – Same as parameter *Attributes* to `create_queue()`

Returns A Queue resource

Return type `SQS.Queue`

get_or_create_queue (*queue_data: Dict[str, str], create_queue: bool = True*)

Fetch or create the sqs Queue resource from boto3.

Also tries to create the queue resource with the configured credentials as dictated by the `create_queue` parameter if the resource was not located.

Parameters

- **queue_data** (*dict[str, str]*) – Dictionary referencing parameters for the queue to be retrieved or created.

The keys for the data are: `name`, `url` and `visibility_timeout`. The `visibility_timeout` defaults to `QUEUE_VISIBILITY_TIMEOUT`.

- **create_queue** (*bool*) – Force creation of queue resource on AWS. Default is `True`

Returns An `Queue` resource on success, `None` otherwise.

Return type `SQS.Queue` or `None`

get_queue (*queue_data*: Dict[str, str])

Retrieve the Queue resource based on provided parameters.

Parameters **queue_data** (*dict*[str, str]) – Same as used for *get_or_create_queue()*

Returns

4.2.3 sqspy.consumer module

class sqspy.consumer.**Consumer** (*queue_name*: Optional[str] = None, *queue_url*: Optional[str] = None, ****kwargs**)

Bases: *sqspy._base.Base*

Message consumer/worker.

Parameters

- **queue_name** (*str*) – Optional queue name.
- **queue_url** (*str*) – Optional queue url, according to AWS guidelines.
- **queue** (*SQS.Queue*) – Optional queue resource.
- **visibility_timeout** (*str*) – Message visibility timeout in seconds, but as a string value. Defaults to *QUEUE_VISIBILITY_TIMEOUT*
- **error_queue** (*str*) – Name for error queue, when messages were not consumed successfully.
- **error_queue_url** (*str*) – Queue url as per AWS guidelines, for the error queue.
- **error_visibility_timeout** (*str*) – Same as *visibility_timeout* but for error queue.
- **create_queue** (*bool*) – Set to *False* if the queue should not be created in case it does not exist. The default is *True*.
- **create_error_queue** (*bool*) – Same as *create_queue* but for error queue. The default is *True*.
- **poll_interval** (*int*) – Polling interval between messages. Defaults to *poll_interval*.
- **message_attribute_names** (*list*[str]) – List of attributes for message to fetch. See *SQS.Message.message_attributes*.
- **wait_time** (*int*) – Time to wait (in seconds) when fetching messages. Defaults to *wait_time*.
- **force_delete** (*bool*) – Whether to delete the message from queue before handling or not. Defaults to *False*.
- **max_messages_count** (*int*) – Maximum message count when fetching from the queue. Defaults to *max_messages_count*.
- **attribute_names** (*list*[str]) – Attributes to be retrieved along with message when fetching. See more at: *SQS.Queue.receive_messages()*

Raises **ValueError** – At least one of *queue*, *queue_url* or *queue_name* has to be provided.

property **error_queue**

The Queue resource for when messages were not processed correctly.

Return type *SQS.Queue*

abstract handle_message (*body, attributes, messages_attributes*)

Method representing the handling of messages retrieved from queue.

Parameters

- **body** (*Any*) – The body retrieved from the queue after passing through a json deserialiser.
- **attributes** (*dict*) – A map of the attributes requested from queue when fetching messages.

See more at: `SQS.Message.attributes`

- **messages_attributes** (*dict*) – Structured metadata as retrieved from the queue.

See more at: `SQS.Message.message_attributes`

Return type `None`

Raises `NotImplementedError` – If not overridden in a subclass.

listen ()

Method that triggers listening for messages, and forwards to `handle_message()`.

This is a blocking call.

max_messages_count: `int = 1`

Upper limit of message count when fetching from queue.

poll_interval: `int = 60`

Time between continuous fetch from queue (in seconds).

poll_messages ()

Poll the queue for new messages.

The polling happens as per the `poll_interval` specified, and the message fetch timeout is set as per the value in `wait_time`.

Returns A list of message resources.

Return type `list[SQS.Message]`

property queue

The connected Queue resource.

Return type `SQS.Queue`

property queue_name

Base name of the connected Queue resource.

Return type `str`

wait_time: `int = 0`

Wait time when fetching message from queue (in seconds).

4.2.4 sqspy.producer module

class sqspy.producer.Producer (*queue_name: Optional[str] = None, queue_url: Optional[str] = None, **kwargs*)

Bases: *sqspy._base.Base*

Message producer.

Parameters

- **queue_name** (*str*) – Optional queue name.
- **queue_url** (*str*) – Optional queue url, according to AWS guidelines.
- **queue** (*SQS.Queue*) – Optional queue resource.
- **visibility_timeout** (*str*) – Message visibility timeout in seconds, but as a string value. Defaults to *QUEUE_VISIBILITY_TIMEOUT*
- **create_queue** (*bool*) – Set to *False* if the queue should not be created in case it does not exist. Default value is *True*.

Raises **ValueError** – At least one of *queue*, *queue_url* or *queue_name* has to be provided.

publish (*message: Any, **kwargs*)

Method to publish message to queue.

The message should be json serializable. The other arguments can be sent as named parameters. More information is available at *SQS.Queue.send_message()*.

Parameters **message** (*Any*) – The message body.

Returns Dictionary of attributes as per AWS guidelines. Check: *send_message()*.

Return type *dict*

property **queue**

See *queue*

property **queue_name**

See *queue_name*

4.2.5 Module contents

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

`sqspy`, 11
`sqspy._base`, 8
`sqspy.consumer`, 9
`sqspy.producer`, 11

B

Base (class in *sqspy._base*), 8

C

Consumer (class in *sqspy.consumer*), 9

create_queue() (*sqspy._base.Base* method), 8

E

error_queue() (*sqspy.consumer.Consumer* property), 9

G

get_or_create_queue() (*sqspy._base.Base* method), 8

get_queue() (*sqspy._base.Base* method), 8

H

handle_message() (*sqspy.consumer.Consumer* method), 9

L

listen() (*sqspy.consumer.Consumer* method), 10

M

max_messages_count (*sqspy.consumer.Consumer* attribute), 10

module

sqspy, 11

sqspy._base, 8

sqspy.consumer, 9

sqspy.producer, 11

P

poll_interval (*sqspy.consumer.Consumer* attribute), 10

poll_messages() (*sqspy.consumer.Consumer* method), 10

Producer (class in *sqspy.producer*), 11

publish() (*sqspy.producer.Producer* method), 11

Q

queue() (*sqspy.consumer.Consumer* property), 10

queue() (*sqspy.producer.Producer* property), 11

queue_name() (*sqspy.consumer.Consumer* property), 10

queue_name() (*sqspy.producer.Producer* property), 11

QUEUE_VISIBILITY_TIMEOUT (*sqspy._base.Base* attribute), 8

S

sqspy

module, 11

sqspy._base

module, 8

sqspy.consumer

module, 9

sqspy.producer

module, 11

W

wait_time (*sqspy.consumer.Consumer* attribute), 10