Henson-AMQP Documentation

Release 0.6.0

iHeartRadio

Jun 25, 2018

Contents

A library for interacting with AMQP with a Henson application.

CHAPTER 1

Installation

Install with pip:

```
$ python -m pip install henson-amqp
```

CHAPTER 2

Quickstart

```
# settings.py
AMQP_INBOUND_QUEUE = 'incoming'
AMQP_INBOUND_EXCHANGE = 'incoming'
AMQP_OUTBOUND_EXCHANGE = 'outgoing'
AMQP_INBOUND_ROUTING_KEY = 'outgoing'
AMQP_OUTBOUND_ROUTING_KEY = 'outgoing'
```

```
# app.py
from henson import Application
from henson_amqp import AMQP

from . import settings
from .callback import run

app = Application('app', callback=run)
app.config.from_object(settings)

amqp = AMQP(app)
app.consumer = amqp.consumer()

# Enable optional Retry support
from henson.contrib.retry import Retry
app.settings['RETRY_CALLBACK'] = app.consumer.retry
Retry(app)
```

```
$ henson run app
```

Contents:

2.1 Settings

2.1.1 Connection Settings

AMQP_HOST	The hostname or IP address of the AMQP server to connect to. Defaults to `'localhost'`.
AMQP_PORT	The port of the AMQP server to connect to. Defaults to `5672`.
AMQP_USERNAME	The username to authenticate with. Defaults to `'guest'`.
AMQP_PASSWORD	The password to authenticate with. Defaults to `'guest'`.
AMQP_VIRTUAL_HOST	The virtual host to use. Defaults to `'/'`.
AMQP_HEARTBEAT_INTERVAL	The heartbeat interval to use for connections. Defaults to `60`.
AMQP_CONNECTION_KWARGS	Additional arguments to pass to `aioamqp.connect()`. Defaults to `{}`.

2.1.2 Consumer Settings

REGISTER_CONSUMER	If `True`, a consumer will be automatically created and assigned to the application. Defaults to `False`.
AMQP_INBOUND_EXCHANGE	The name of the exchange that the consumer should read from. Defaults to `''` (the AMQP default exchange).
AMQP_INBOUND_EXCHANGE_DURABLE	The durability setting of the exchange that the consumer reads from. Defaults to `False`.
AMQP_INBOUND_EXCHANGE_TYPE	The type of the inbound exchange. Defaults to `'direct'`.
AMQP_INBOUND_EXCHANGE_KWARGS	Additional arguments to pass to `aioamqp.channel.exchange_declare()`. Defaults to `{}`.
AMQP_INBOUND_QUEUE	The name of the queue that the consumer should read from. Defaults to `''` (the AMQP default queue).
AMQP_INBOUND_QUEUE_DURABLE	The durability setting of the queue the consumer reads from. Defaults to `False`.
AMQP_INBOUND_ROUTING_KEY	The routing key used to bind the inbound exchange and queue. Defaults to `''`.
AMQP_DISPATCH_METHOD	Reserved for future use.

2.1.3 Producer Settings

AMQP_OUTBOUND_EXCHANGE	The name of the exchange used by the producer to send messages. Defaults to `''`.
AMQP_OUTBOUND_EXCHANGE_DURABLE	The durability setting of the outbound exchange. Defaults to `False`.
AMQP_OUTBOUND_EXCHANGE_TYPE	The type of the outbound exchange. Defaults to `'direct'`.
AMQP_OUTBOUND_EXCHANGE_KWARGS	Additional arguments to pass to `aioamqp.channel.exchange_declare()`. Defaults to `{}`.
AMQP_OUTBOUND_ROUTING_KEY	The default routing key used when sending messages to the outbound exchange if the `routing_key` argument is not provided. Defaults to `''`.
AMQP_PREFETCH_LIMIT	The maximum number of messages to keep in the internal queue waiting to be processed. If set to `0`, the consumer will fetch all available messages from the AMQP queue. Defaults to `0`.
AMQP_DELIVERY_MODE	The mode used when sending messages. By default, messages are non-persistent. Defaults to *henson_amqp.DeliveryMode.NONPERSISTENT*

2.2 API

The public API of Henson-AMQP.

2.2.1 AMQP

class henson_amqp.**AMQP**(*app=None*)
An interface to interact with an AMQP broker.

consumer()
Return a new AMQP consumer.

Returns

A new consumer object that can be used to read from the AMQP broker and queue specified the Application's settings.

Return type *Consumer*

init_app(*app*)
Initialize the application.

If the application's REGISTER_CONSUMER setting is truthy, create a consumer and attach it to the application.

Parameters **app** (*henson.base.Application*) – The application instance that will be initialized.

producer()
Return an AMQP producer, creating it if necessary.

Returns

A new producer object that can be used to write to the AMQP broker and exchange specified by the Application's settings.

Return type *Producer*

2.2.2 Consumer

class henson_amqp.**Consumer**(*app*)
A consumer of an AMQP queue.

Parameters **app** (*henson.base.Application*) – The application for which this consumer consumes.

read()
Read a single message from the message queue.

If the consumer has not yet begun reading from the AMQP broker, that process is initiated before yielding from the queue.

Returns The next available message.

Return type *Message*

Raises aioamqp.exceptions.AioamqpException – The exception raised on connection close.

retry(*app*, *message*)
Requeue a message to be processed again.

This coroutine is meant for use with the `henson.contrib.retry.Retry` extension.

Parameters

- **app** (*henson.base.Application*) – The application processing the message.
- **message** (*dict*) – A copy of the message read from the AMQP server.

Note: This function assumes that messages are JSON serializeable. If they are not, a custom function may be used in its place.

2.2.3 Message

class henson_amqp.**Message**(*body*, *envelope*, *properties*)

body
Alias for field number 0

envelope
Alias for field number 1

properties
Alias for field number 2

2.2.4 Producer

class henson_amqp.**Producer**(*app*)
A producer of an AMQP queue.

Parameters **app** (*henson.base.Application*) – The application for which this producer produces.

send(*message*, *, *routing_key=None*)
Send a message to the configured AMQP broker and exchange.

Parameters

- **message** (*str*) – The body of the message to send.
- **routing_key** (*str*) – The routing key that should be used to send the message. If set to `None`, the `AMQP_OUTBOUND_ROUTING_KEY` application setting will be used. Defaults to `None`.

2.2.5 DeliveryMode

class henson_amqp.**DeliveryMode**
AMQP message delivery modes.

NONPERSISTENT = 1
Mark messages as non-persistent before sending to the AMQP instance.

PERSISTENT = 2
Mark messages as persistent before sending to the AMQP instance.

2.3 Changelog

2.3.1 Version 0.6.0

Released 2018-06-25

- Support both `int` values and existing `henson_amqp.DeliveryMode` `enums` for the `AMQP_DELIVERY_MODE` setting
- Stop declaring the exchange with every message sent

2.3.2 Version 0.5.0

Released 2016-06-24

- Support passing `arguments` to inbound and outbound exchange declaration

2.3.3 Version 0.4.0

Released 2016-04-27

- Allow callers of `Producer.send` to specify a routing key

2.3.4 Version 0.3.0

Released 2016-04-08

- Add support for `Retry`
- Add `REGISTER_CONSUMER` setting

2.3.5 Version 0.2.0

Released 2016-03-11

- If a connection is closed by `aioamqp` while reading, raise that exception during calls to `Consumer.read` after all previously read messages have been returned.

2.3.6 Version 0.1.0

Released 2016-03-01

- Initial release

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Index