classymq Documentation

Release 0.7.0

Greg Doermann

October 02, 2014

Contents

classymq is a class based RabbitMQ library.

CHAPTER 1

Download

Github

PyPi

CHAPTER 2

Contents:

2.1 Frequently Asked Questions

2.1.1 Q: Why use a class based system?

A: classymq allows you to simply declare your interfaces and then all the connection, messaging, cleaning, declaring, etc happens for you and you just create and receive message classes.

CHAPTER 3

Getting Started

```
#first, define an exchange
from classymq import exchanges
class MyExchange(exchanges.BaseExchange):
    KEY = "myexchange"
    TYPE = exchanges.EXCHANGE_TYPES.TOPIC
    NOWAIT = True
    DURABLE = False

#second, setup the queue
from classymq import queues
class MyQueue(queues.BaseQueue):
    KEY = "myqueue"
    NOWAIT = True
    DURABLE = False

#third, create JSON serializable message definition class
class MyMessage(AttrDict):
    def __init__(self, message, uuid=None):
        AttrDict.__init__(self)
        self.message = message
        self.uuid = uuid

#fourth, create a consumer and register it to do something on incoming messages
from classymq import consumers
class MyConsumer(consumers.JsonMessageConsumer):
    EXCHANGE = MyExchange
    QUEUE = MyQueue
    ROUTING_KEY = '#'
    MESSAGE_CLASS = MyMessage

    def __init__(self, key_params=None, rate=None, routing_keys=None):
        super(MyConsumer, self).__init__(key_params, rate, routing_keys)
        self.processor.register(self.do_something)

    def do_something(self, message):
        print 'recieved message {}: {}'.format(message.uuid, message.message)

#fifth, create a producer class
from classymq import producers, synchronous

# twisted producer
class MyProducer(producers.JsonProducer):
    CONSUMER = MyConsumer
```

```
# synchronous producer (pika)
class MySynchronousProducer(synchronous.JsonProducer):
    CONSUMER = MyConsumer


#next, start listening on the consumer side
# myservice.py
from twisted.internet.defer import inlineCallbacks
from twisted.internet import reactor
from classymq import factory, settings
import MyConsumer

consumer = MyConsumer()

@inlineCallbacks
def run():
    factory = AmqpFactory()
    yield factory.connect()
    factories.append(factory)
    factory.read(consumer)

run()
reactor.run()


#finally, produce a message
>>> import MySynchronousProducer, MyMessage, uuid
>>> producer = MySynchronousProducer()
>>> message = MyMessage("Hello world!", str(uuid.uuid4())
>>> producer.push(message)
```

Report a Bug

Users Mailing List

CHAPTER 4

Indices and tables

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