
kombu_redis_priority Documentation

Release 0.0.7

Shaon Barman, AJ Renold, Yoriyasu Yano

February 03, 2017

Contents:

1	Installation	3
2	Testing	5
3	Indices and tables	7

kombu_redis_priority is a plugin into kombu that provides a Redis backed transport backend that supports prioritized ordering of messages through SortedSet.

kombu_redis_priority works by extending the existing redis transport backend and replacing references to the list data type with sortedset. It then utilizes the priority interface provided in AMQP to order the messages within the sortedset, thereby allowing implementations of prioritized messaging.

In addition, it takes advantage of lexicographical ordering of messages to achieve FIFO queueing when the priorities are equivalent. This is achieved by prefixing message jsons with epoch timestamps at the time of enqueueing messages into the sortedset.

Installation

To get started using the `redis_priority` transport:

First, install the package:

```
pip install kombu-redis-priority
```

Then, import the package at the start of your application, before you start configuring kombu. For example in a Celery application, you would add the following line before you configure your celery application in a `celery.py` file:

```
import kombu_redis_priority.transport.redis_priority_async
app = Celery('redis_priority_example')
```

You can now use the `redis_priority` transport by referring to the *redispriority* transport wherever you configure kombu.

Testing

kombu_redis_priority uses [Travis](#) for CI builds and [coveralls](#) for coverage reporting.

You can also run the tests locally using `setuptools`:

```
python setup.py test
```

Indices and tables

- `genindex`
- `modindex`
- `search`