django-rq-mail Documentation

Release 0.1.3.alpha

Oct 04, 2017

Contents

This project is not maintained anymore, it doesn't support latest changes from rq

django-rq-mail is a simple Python library based on rq to store emails sent by Django and process them in the background with workers.

As django-rq-mail is based on rq, it's entirely backed by Redis.

CHAPTER 1

Architecture

django-rq-mail adds new elements to enjoy Sorted Sets from Redis.

For the purpose of django-rq-mail, it implements the concept of `WaitingQueue` which delays the processing of a job with a timestamp.

The default behavior of rq is to process jobs via BLPOP which blocks the connection when there are no elements to pop from any of the given queues. With this behavior there is no way to delays the processing of a job and when it's failing rq pushs it in a failed queue. Of course, you can requeue this job later but there is no fallback mechanism.

In django-rq-mail you can define fallback steps (in seconds) to retry a job until it's not failing. When a job has been tested on each steps we reintroduce the default behavior of rq on pushing it in the failed queue.

Each steps will create a waiting queue and when a job is failing we take the current timestamp with the delta to retry it in the future.

This mechanism is possible with ZADD which adds a serialized job in the queue with a score and ZREVRANGEBYSCORE to return all the elements in the sorted set with a score between max (current timestamp) and min.

As you may understood, we have dropped the default blocking behavior to replace it by a daemon which is running each seconds.

CHAPTER 2

Installation

1. Either check out the package from GitHub or it pull from a release via PyPI

```
pip install django-rq-mail
```

2. Add 'rq_mail' to your `INSTALLED_APPS`

```
INSTALLED_APPS = (
    'rq_mail',
)
```

to use the *rq_mail* command (via Django commandline) shipped by django-rq-mail.

This command is a minimal integration of rq into Django to launch the **Dispatcher**.

3. Define `EMAIL_BACKEND`

```
EMAIL_BACKEND = 'rq_mail.backends.RqBackend'
```

4. Define `RQ_MAIL_EMAIL_BACKEND` the backend used to send your emails, for example

```
RQ_MAIL_EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
```

CHAPTER 3

Logging

RQ 0.3.3 uses standard Python's `logging`, this means you can easily configure `rqworker`'s logging mechanism in django's `settings.py`. For example:

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'formatters': {
        'rq_console': {
            'format': '%(asctime)s %(message)s',
            'datefmt': '%H:%M:%S',
        },
    },
    'handlers': {
        'rq_console': {
            'level': 'DEBUG',
            'class': 'rq.utils.ColorizingStreamHandler',
            'formatter': 'rq_console',
            'exclude': ['%(asctime)s'],
        },
    },
    'loggers': {
        'rq.worker': {
            'handlers': ['rq_console'],
            'level': 'DEBUG'
        },
    }
}
```

CHAPTER 4

Utilisation

Once you have installed it, you can run `python manage.py rq_mail` from your shell.

CHAPTER 5

Configuration

RQ_MAIL_PREFIX

The prefix used to name all queues created by django-rq-mail.

RQ_MAIL_MAIN_QUEUE

The name of the main queue.

RQ_MAIL_EMAIL_BACKEND

The email backend used to send emails when they are processed in the background.

RQ_MAIL_REDIS_HOST

The Redis host used to connect.

RQ_MAIL_REDIS_PORT

The Redis port used to connect.

RQ_MAIL_REDIS_DB

The Redis database used to connect.

RQ_MAIL_REDIS_PASSWORD

The Redis password used to connect.

RQ_MAIL_REDIS_URL

The Redis url used to connect.

RQ_MAIL_REDIS_SOCKET

The Redis socket used to connect.

RQ_MAIL_FALLBACK_STEPS

A simple list of timing to create waiting queues.

You can define as much steps as you want, each will be transformed to a queue. So if you define 10 steps, you will allow a message to fail 10 times until it will go in the failed queue.

CHAPTER 6

Reference

For further details see the reference documentation:

- genindex
- modindex
- search

CHAPTER 7

Issues

For any bug reports and feature requests, please use the Github issue tracker.