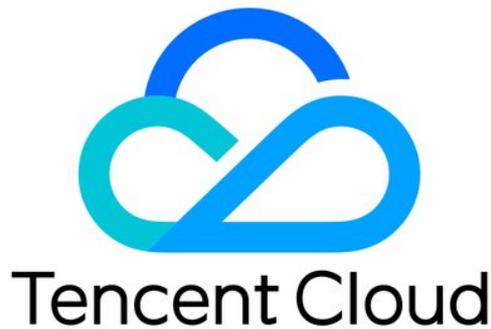


TDMQ for CMQ

Glossary

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Glossary

Last updated : 2024-01-03 10:21:21

Inactive Message

Inactive message refers to the `InactiveMessages` attribute, i.e., the total number of messages with the inactive status in the queue (approximate value).

Long Polling

Long polling refers to a mechanism where when the client sends a request to the server, the server will hold the connection when receiving the request and return a response message and close the connection until a new message arrives, and the client will send a new request to the server after processing the response message.

A message consumption request will return a response only after a valid message is fetched or the long-polling time elapses, which can avoid repeated invalid polling.

Retry Policy

Retry policy refers to the `NotifyStrategy` attribute of a subscription in CMQ, i.e., the policy used to perform retry when an error occurs during message push to receivers.

The retry policy is enabled by default, and you must select one of the following two options:

Backoff retry: An attempt will be retried three times at random intervals between 10 and 20 seconds. After three retries, the message will be discarded for the subscriber and will not be retried again.

Exponential decay retry: An attempt will be retried 176 times for one day at exponentially increasing intervals: 2^0 seconds, 2^1 seconds, ..., 512 seconds, 512 seconds, ..., 512 seconds. This is the default retry policy.

Retry Verification

For retry verification, after a message is delivered to a subscriber, an HTTPS return code of 200 indicates a success.

Subscriber

A subscriber refers to a subscriber to a service in topic mode in CMQ.

Queue

In CI, when you activate the MPS service, the system will automatically create a user queue (queue-1) for you. When you submit a task, the task will be arranged in the queue first and executed in sequence according to the priority and order of submission.

In GSE, a queue is a group of server fleets running in regions of a game asset package. It can be used to create a cross-region fleet group, allowing you to place game server sessions in any fleet in the queue to minimize latency for a better player experience. It also enables you to make the most of your fleet capacity for new games in order to achieve higher availability.

In TDMQ for CMQ, a queue is the destination of message storage, from which consumers actively get messages. In a queue, `MessageId` or `ReceiptHandle` is used to uniquely identify messages.

Active Message

Active message refers to the `Activemessages` attribute, i.e., the total number of messages with the active status in the queue (approximate value).

Polling Wait Time

Polling wait time (`PollingWaitSeconds`) refers to the maximum wait time for a polling to time out in seconds. Its value ranges from 0 to 30 seconds.

Request

A request refers to the content sent by a consumer to a queue in order to get a message.

Hidden Duration of Fetched Message

Hidden duration of fetched message (`VisibilityTimeout`) refers to the duration after which a message will be sent to and processed by another process if it fails to be processed after being received. It will be counted in seconds immediately after a message is received, and its value ranges from 1 second to 43,200 seconds (i.e., 1 second–12 hours).

Production

Production (`Produce`) refers to an operation of writing a message into a topic by a CMQ producer.

Producer

A producer refers to a role that sends messages.

Dead Letter Queue

A dead letter queue is used to process messages that fail to be processed properly. After it is enabled, undeleted messages with consumption times that exceed the limit and expired messages will be delivered to the dead letter queue according to applicable rules.

Consumer

A consumer refers to a role that receives messages.

Message ID

A message ID is the unique identifier of a message, which consists of four parts: `ledgerid` , `entryid` , `partitionindex` , and `batchid` .

Each message will receive a message ID assigned by the Tencent Cloud system, which can be returned to you through the `SendMessage` API request.

Message

In TDMQ for CMQ, a message refers to the content delivered between different processes, which contains data and attributes.

In App Flow, a message refers to the data structure passed between each connector instance and processor instance in the integration flow.

Message Retention

Message retention (`MessageRetentionPeriod`) refers to a process where messages that are produced by a producer but have not triggered delivery to subscribers or failed to be received by subscribers will be retained in the topic temporarily and be delivered again for multiple times. It is enabled by default and is not configurable. The maximum retention period is 1 day.

Message Filter Tag

TDMQ for CMQ allows consumers to filter messages by tags, which ensures that they only consume messages that are relevant to them. After a tag is added, subscribers will receive only messages with the tag. If no tag is set, all messages will be sent to all subscribers.

Message Receipt Mode

Message receipt mode refers to the method for a consumer to get messages. Currently, only the pull mode is supported, that is, the consumer actively gets messages.

Message Receipt Mode (Push)

The message receipt mode (push) is the topic model of TDMQ for CMQ, which already supports active pushes.

Message Handler

A message handler (`ReceiptHandle`) marks a message as manipulatable. Every time you read a message from a message receipt queue, you will receive a handler (`Handle`) that can be used to manipulate the message at the same time.

You can use a message handler to delete or modify some attributes of a message. It is relevant to the message receipt operation rather than the message itself. To delete or modify a message, the message handler instead of the message ID must be provided, which means that a message can be modified/deleted only after it is received.

Message handlers provided by CMQ have a validity period and will expire after the preset duration (which is 30 seconds by default and can be customized) elapses. This effectively avoids faulty data operations and greatly reduces the risks that may be caused by handler disclosure.

Message Content

Message content (`Message Body`) refers to the received message body. The default encoding format of messages received and sent in Tencent Cloud is Base64, which is the same as that of the official SDK of Message Service.

Message Lifecycle

Message lifecycle (`msgRetentionSeconds`) refers to the maximum period of time during which a message can be retained. After the period specified by this parameter has elapsed since a message is sent to the queue, the message will be deleted no matter whether it has been fetched. It is measured in seconds.

In a queue, its value ranges from 60 to 1,296,000 seconds (i.e., 1 minute–15 days).

In a topic, it is 86,400 seconds (1 day) by default and cannot be modified.

First Consumption Time of Message

The first consumption time of a message (`FirstDequeueTime`) refers to the time when a message in a queue is consumed for the first time.

Next Consumption Time of Message

The next consumption time of a message (`NextVisibleTime`) refers to the time when a received message can be consumed again.

Message Consumption Count

Message consumption count (`DequeueCount`) refers to the total number of times that a message is consumed in a queue.

Message Digest

Message digest (`MsgBodyMD5`) is a fixed-length value uniquely corresponding to a message or piece of text, which is used to check whether the information is tampered with during consumption.

Maximum Message Length

Maximum message length (`MaxMsgSize`) refers to the maximum length of a message body that can be sent to a queue. It is measured in bytes and ranges from 1,024 to 65,536 bytes (i.e., 1–64 KB).

Topic

In Oceanus, a topic is the smallest unit of stream connection to subscription and publishing. You can use a topic to denote a type of streaming data, just like a table in a database.

In IoT Hub, a topic refers to a message communication topic, acting as the communication intermediary of the messages in the Pub/Sub pattern. It is required when publishing and subscribing to messages, and the communication is made based on the specific topic of each device.

In TDMQ, CKafka, and CMQ, a topic refers to a collection of messages of a particular type. It is a logical concept used to store messages. Topics must be unique in a namespace.

In CLS, a topic refers to the basic management that corresponds to an application or service. It is the smallest management unit for log collection, indexing, and delivery. There can be multiple log topics in a logset.

In BI, a topic is a way to batch modify the styles of the overall components in order to make the report design more uniform. In BI systems, topics are often used for unified settings of chart components and page backgrounds.