

# Message Service

## Product Introduction

# Product Introduction

Alibaba Cloud Message Service is a high performance, reliable, safe, extensible distributed message and notification service that supports massive messages, concurrent operations, which facilitates message transfer between applications and system decoupling.

Message Service provides 2 models:

## Queue

Designed to send or receive the message in **Point-to-Point** way. The consumer receives a message by pulling it down.

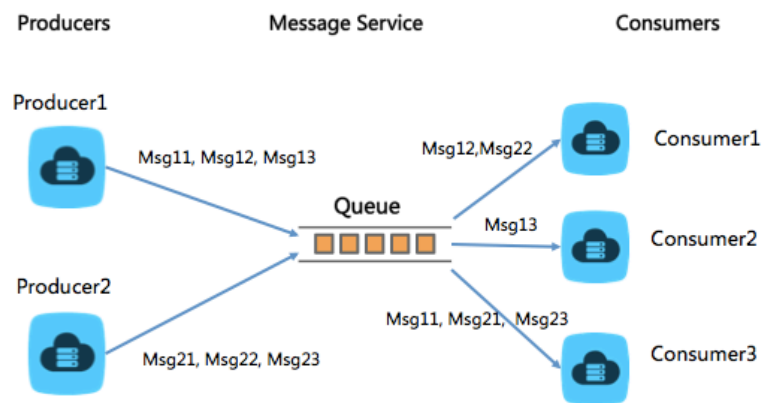
## Topic

Designed to send or receive the message in **Pub/Sub** way. The message is pushed from message broker to consumers. The consumers can be http, queue, SMS, Email, mobile phone.

## Overview for Queue

Queue is designed to provide consuming message in point-to-point way for high concurrent scenario. Each message can be consumed only once.

### Picture for queue model



## Characteristics of Queue model

### A. Frutful attributes

Queue supports fruitful customizable attributes to meet the requirements of different scenarios, such as normal queue, delay queue, priority queue.

### B. Support massive request concurrently

One queue can be accessed by many produders or consumers at the same time while one message can not be fetched by other consumers within a certain period of time after being fetched by one consumer.

### C.Ensurance for message delivery

The message which is sent to the queue successfully is guaranteed to be fetched at least once and prevent the illegal access from others.

### D. Support transaction message

It provides function to rollback the message sent to the queue which help application do transaction rollback.

### E.Support logging for operations on message.

It provides log function to query the operations of sending, receiving and deleting on messages. It' s helpful while debugging.

### F. Support Alibaba Cloud Watch

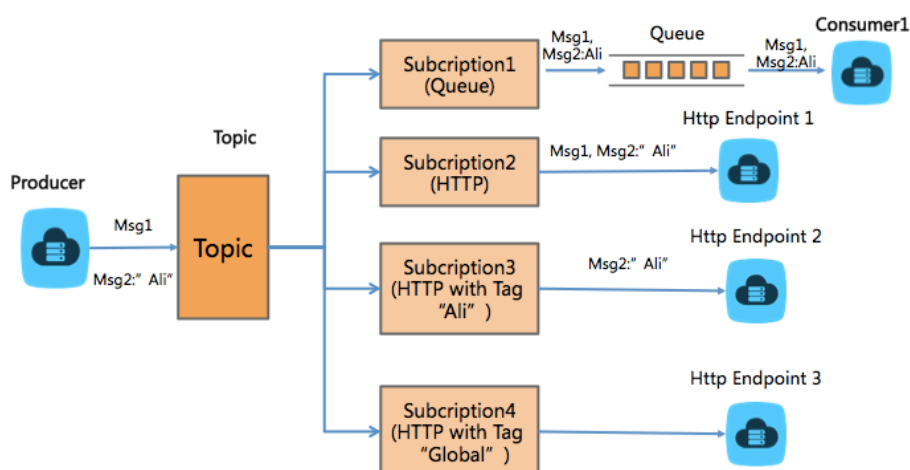
User can get information of queue from CloudWatch and customize the alert.

## Overview for Topic

Topic is notification function in pub/sub model which broadcasts messages from one publisher to multiple subscribers. It supports several ways for message notification:

- Push the message to http server specified by user.
- Push the message to a MNS queue.
- Push the message to an E-Mail address or E-Mail address group.
- Push the message to the phone number by SMS(China only now).
- Push the message to the Browser by WebSocket(in planning).
- Push the message to the mobile device.

### Picture for Topic :



### Characteristics for Topic

- Support notification message.
- Support one-to-multiple message.
- Support filter tag for message.
- Support different ways to push message.
- Ensurance for message delivery.
- Support for event message from Alibaba Cloud services.
- Support logging for operations on message.
- Support Alibaba Cloud Watch.

#### A. Support notification of message

Message Service server-side pushes messages to the http url which specified by the users. In this way, the applications of users have no need to handle the message broker server any more, which facilitates the users to save their resource such as CPU, focus on their logic and reduce the dependency on the Message Service SDK.

Message service server-side pushes messages from topic to queue when the users are

inconvenient to provide the http url due to the network problem, which make applications consume messages from queue in time.

## **B. Support one-to-multiple message.**

A message sent to topic can be subscribed by multiple consumers and pushed to mutiple endpoints in the way designated by the consumers.

## **C. Support filter tag for message**

Tag(at most of time , it' s a string) can be specified while doing subscription on topic for endpoints. Also tag can be put as attribute of the message while publishing. Topic only sends the message to endpoint with same tag(string). The subscription of endpoint not specified with tag will receive all of the messages published to the topic. And the message not set tag, will be sent to all the endpoints.

## **D. Support different ways to push message**

The ways below are supported:

- Push the message to the http server specified by users. Details : HttpEndpoint , HttpEndpoint Process , HttpEndpoint in Java , HttpEndpoint Local Debug Tool.
- Push the message to a MNS queue.
- push the message to an E-Mail address or E-Mail address group.
- Push the message to phone number by SMS.
- Push the message to Browser by WebSocket(in planning).
- Push the message to mobile device.

## **E. Ensurance for message delivery**

In the lifecycle of message, the message is pushed in the strategy and format specified by user. The push strategies including:

- BACKOFF\_RETRY  
Retry 3 times in total (with the interval of 10 to 20 seconds) for each message.
- EXPONENTIAL\_DECAY\_RETRY  
Retry 176 times in total for each message. The intervals between adjacent retries are sequential as following order: 1, 2, 4, 8 , 16, 32, 64, 128, 256, 512, 512, 512... The maximum time interval is 512 seconds.

The formats of message including: :

- XML : Message body is in XML format with message attributes;
- JSON : Message body is in JSON format with message attributes;
- SIMPLIFIED : message body is raw text;

## **F. Support for event message from Alibaba Cloud services.**

Message service provides event/notification function to subscribe event from the cloud resources of other Alibaba Cloud Service such as OSS(Object Storage Service), DM(DirectMail) Service, SLS(Simple Log Service) etc. This function enable users to set rules for the event of operations on OSS object including: CreateObject, DeleteObject, UpdateObject etc.

## **G. Support logging for operations on message.**

Every operation on message is logged down to the SLS or OSS. Users can query the message log with message id in SLS.

## **H. Support Alibaba Cloud Watch.**

Users can get information of topic from CloudWatch and customize the alert.

# **Queue Concepts**

Queue is the destination storage of messages. Message consists of data and attributes. MessageId and ReceiptHandle are two types of indicator for an unique message in the queue.

## **Account**

The user ID of Alibaba Cloud Service.

## **Owner of Queue**

The Account opened with Message Service uses CreateQueue API to create a queue, then the Account is the owner of the queue. The owner has all the rights of operations on the queue. The AccountID can be queried from the Alibaba Cloud WebSite.

## **Producer**

The role of sending message to queue.

## **Consumer**

The role of receiving message and deleting message from queue.

## **Endpoint**

The domain url to access Message Service in following format: "  
**`https://$AccountId.mns.$Region.aliyuncs.com`**"

\$Region is the location where Message Service deployed.

\$AccountId is the user ID given by Alibaba Cloud Service when user register to use Alibaba Cloud Service.

## Normal Message

The message sent to queue can be received by consumer at once.

## Delay Message

The message sent to queue will be hidden some time before available for receiving by consumer.

## Normal Queue

The default message delay time is zero which means any message not set with delay seconds attribute is available for receiving at once.

## Delay Queue

The default message delay time is bigger than zero, which means any message not set with delay seconds attribute will be delayed by specified seconds before visible to the consumers.

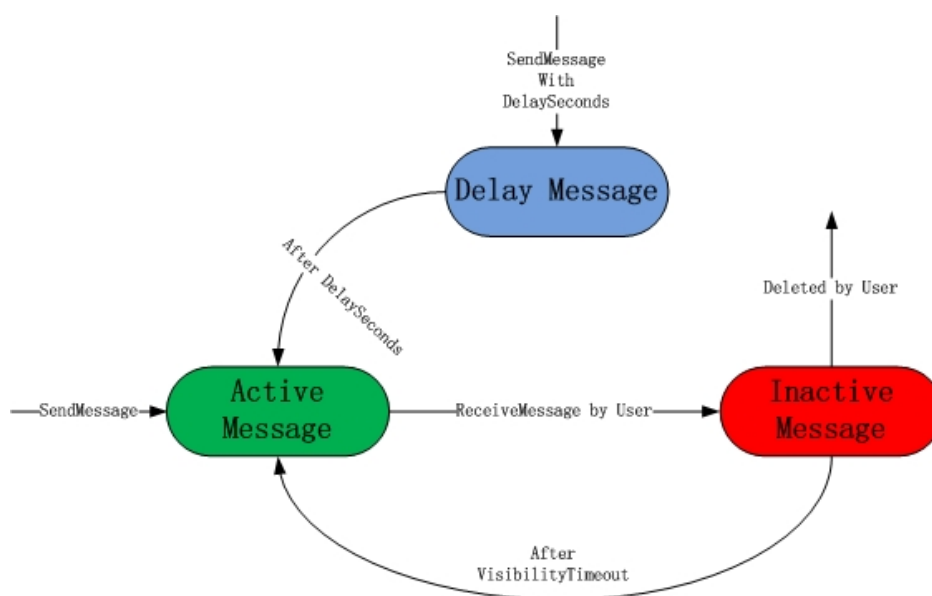
## MessageID

The ID indicating a message is generated by Message Service when producer sends message request to the queue. Every message has a unique MessageID in the same queue. But the MessageID may be duplicated in different queues. And the MessageID can return the response from Message Service but can not be used to delete a message.

## ReceiptHandle

ReceiptHandle is a temporary indicator. It is generated by Message Service and returned to consumers when they receive messages. Consumers can use the ReceiptHandle to delete message or change visibility timeout of message. But the ReceiptHandle will expire if the message status changes.

## Message Status



## Message State Transition Diagram

There are three statuses for message in queue: **Active, Delayed, Inactive**.

Normal message is initialized with status Active. Delay message is initialized with status Delayed.

If the total time of the message stay on Delayed status reaches the delay time, it will transit into Active status.

If message in Active status is received by a consumer, the message status will transit to Inactive status and is not visible for consumers.

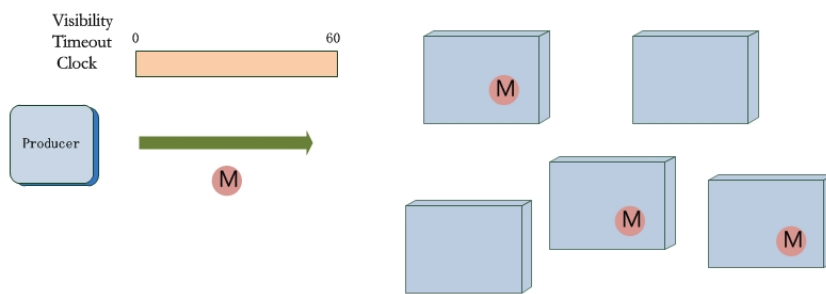
There is a visibility timeout for every Inactive message. If the Inactive message is not deleted by the consumer within the visibility timeout, the status will transit to Active and can be received by consumers again.

Both the message delay time and visibility timeout can be specified by user application.

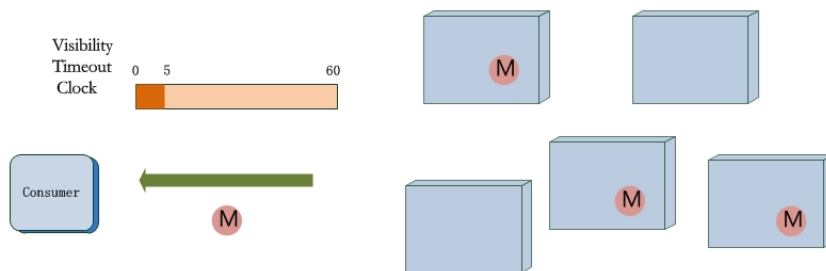
## Message Life Cycle



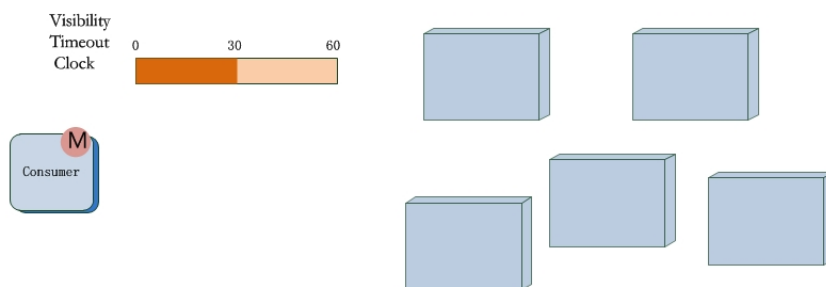
**1** The producer sends message "M" to a queue.



**2** The consumer obtains message "M" from the queue, and the VisibilityTimeoutClock starts timing.



**3** After processing message "M", the consumer deletes it from the queue before the VisibilityTimeoutClock expires.



## Topic

Topic is the destination and the storage address of message in the message Pub/Sub model.

## Account

The user id of Alibaba Cloud which consists of a sequence of numbers.

## Publisher

The client sending message to the topic, it's also known as "producer".

## Subscriber

The client receives the message from topic. The owner of topic has authorization of subscribing the owned topic.

## Subscription

The relationship between message subscriber and topic. The message which publisher sends to topic can only be received by subscribers.

## Endpoint

The address that the subscriber used to receive messages, such as http address, email address and phone number(for SMS) etc.

## TopicURL

Indicator in format: "http(s)://\$AccountId.mns.\$Region.aliyuncs.com/topics/\$TopicName" in which \$AccountId means the user account id(uid), \$Region is the name of region where AliCloud Message Service provides service, \$TopicName is the name of topic.

## Message

The unit of information transferred from publisher to subscriber through topic. There is an unique MessageID for every message in the same topic but may be duplicated in different topics.

Every message will be kept for 1 day after sending to topic. The message existing over 1 day will expire and be recycled soon.

## The Advantages of Message Service comparing with Self-Build Queue Service:

Key Advantages	Using Message Service	Using Self-Build Queue Service
Easy-to-Use	Need zero cluster and zero operations; Standard HTTP REST API; Fruitful SDKs for most popular programming languages.	Need cluster , high cost at start time; Need operations and cost increase while business growing up; Private protocol instead of Standard HTTP REST API.
Stability	Store 3 copies for messages, data security is 99.99999999%; Service availability is 99.9% ; Ensurance for delivery message successfully at once;	Message is stored on single or two machines; Small cluster without high availability;
Security	Professional protection in	Nearly no security strategies;

	different levels to prevent from attacks such as DDOS; Isolation and dedicated namespace for multiple users; Support Main-Sub accounts for authorization and authentication; Support HTTPS and VPC;	
Scalability	The count of queue and the storage of message are not limited; Auto scale up transparent to users; Deployed worldwide(13 regions now);	Limited by count of queue, count of messages due to storage limitation; Limited by region; Not scale up.

## Other Advantages:

## High Performance Price Ratio

- Low price with high performance ;
- Discount while buying the resource package ;

## Professional Technique Support

- 24/7 technique support for worksheet ;
- One-to-One technique support service on demand(In China Area) ;
- One-to-One technique support IM (Worldwide);
- Product forum for questions(forum link).