

# **TDMQ for RocketMQ**

## **Getting Started**

### **Product Documentation**



## Copyright Notice

©2013-2019 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.

# Contents

## Getting Started

- Resource Creation and Preparation

- Downloading and Running Demo

# Getting Started

## Resource Creation and Preparation

Last updated : 2022-07-04 16:38:20

### Overview

This document describes how to create resources such as cluster and topic in the TDMQ console and what operations you need to perform in the console before running a client.



### Prerequisites

- You have [signed up for a Tencent Cloud account](#).

### Directions

#### Step 1. Create a cluster



- Log in to the [TDMQ console](#), enter the **Cluster** page, and select the target region.
- Click **Create Cluster** and enter the cluster name and description to create a cluster.

<input type="checkbox"/> Cluster ID/Name	Topics	Group Count	Cluster Description	Resource Tag	Operation
<input type="checkbox"/> rocketmq-  /tz8 test	Used: 1 Capacity: 1000	Used: 1 Capacity: 10000		 3	<a href="#">Access Address</a> <a href="#">Edit</a> <a href="#">Delete</a>

- On the **Cluster** list page, click **Access Address** in the **Operation** column of the cluster you just created to get the connection information of the server.

## API Call Address

### VPC Access Address

rocketmq- kvmmz8.rocketmq.ap-  
sh.qcloud.tencenttdmq.com:5022 

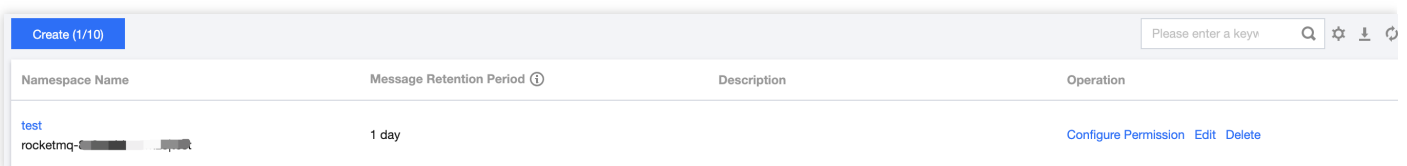
### Public Network Access Address



rocketmq- kvmmz8.rocketmq.ap-  
sh.public.tencenttdmq.com:9876 

OK

## Step 2. Create a namespace

1. On the **Cluster** list page, click the ID of the cluster created in **step 1** to enter the cluster's basic information page.
2. Select the **Namespace** tab at the top, click **Create**, and set the namespace name, message retention period, and description to create a namespace.



Namespace Name	Message Retention Period 	Description	Operation
test rocketmq- 	1 day		<a href="#">Configure Permission</a> <a href="#">Edit</a> <a href="#">Delete</a>

## Step 3. Create a role and configure permissions

1. Select **Role Management** on the left sidebar and click **Create** to create a role.
2. On the **Cluster** page, click the ID of the cluster you just created to enter the cluster details page.
3. Select the **Namespace** tab at the top and click **Configure Permissions** in the **Operation** column of the namespace you just created.

- On the **Configure Permission** page, click **Add Role** to add production and consumption permissions to the role you just created.

## Create ✕

Role bigdata ▼

Unable to find a role? Please configure a role and key on the [Role Management](#) ↗ page.

Permission  Message production  
 Message consumption

For more permission type information, see [here](#) ↗

Save
Cancel

## Step 4. Create a topic

- On the **Namespace** list page, select the **Topic** tab at the top to enter the **Topic** list page.
- Select the namespace created in [step 3](#) and click **Create** to create a topic.

Topic Name	Monitori...	Type ▾	Subscribed Groups	Description	Operation
test2		General	0		<a href="#">Edit</a> <a href="#">Delete</a>

## Step 5. Create a group

- On the **Topic** list page, select the **Group** tab at the top to enter the **Group** list page.
- Select the namespace you just created and click **Create** to create a group.

Group Name	Consumer Info ↕	Consumption Mode	Description	Operation
▶ t...t	Online Consumer 0 TPS 0 Total Heap 0	Unknown		<a href="#">Consumer Details</a> <a href="#">Reset Offset</a> <a href="#">Edit</a> <a href="#">Delete</a>

# Downloading and Running Demo

Last updated : 2022-09-06 16:55:59

## Overview

This document describes how to use open-source SDK to send and receive messages by using the SDK for Java as an example and helps you better understand the message sending and receiving processes.

### Note

The following takes the Java client as an example. For clients in other languages, see [TDMQ for RocketMQ](#).

## Prerequisites

- [You have created the required resources.](#)
- [You have installed JDK 1.8 or later.](#)
- [You have installed Maven 2.5 or later.](#)
- [You have downloaded the demo.](#)

## Directions

### Step 1. Install the Java dependency library

Introduce dependencies in a Java project and add the following dependencies to the `pom.xml` file. This document uses a Maven project as an example.

### Note :

The dependency version must be v4.6.1 or later.

```
<!-- in your <dependencies> block -->
<dependency>
<groupid>org.apache.rocketmq</groupid>
<artifactid>rocketmq-client</artifactid>
```




```
<version>4.6.1</version>
</dependency>

<dependency>
<groupid>org.apache.rocketmq</groupid>
<artifactid>rocketmq-acl</artifactid>
<version>4.6.1</version>
</dependency>
```

## Step 2. Produce messages

### 1. Create a message producer

```
// Instantiate the message producer
DefaultMQProducer producer = new DefaultMQProducer(
namespace,
groupName,
new AclClientRPCHook(new SessionCredentials(accessKey, secretKey)) // ACL permis
sion
);
// Set the NameServer address
producer.setNamesrvAddr(nameserver);
// Start the producer instances
producer.start();
```

Parameter	Description														
namespace	Namespace name, which can be copied under the <b>Namespace</b> tab on the cluster details page in the console. Its format is <b>cluster ID +   + namespace</b> .														
groupName	Producer group name, which can be copied under the <b>Group</b> tab on the cluster details page in the console.														
nameserver	Cluster access address, which can be copied from <b>Access Address</b> in the <b>Operation</b> column on the <b>Cluster</b> page in the console.														
secretKey	Role name, which can be copied on the <a href="#">Role Management</a> page.														
accessKey	Role token, which can be copied in the <b>Token</b> column on the <a href="#">Role Management</a> page. <div data-bbox="304 1809 1489 1912" data-label="Table"> <table border="1"> <thead> <tr> <th><input type="checkbox"/></th> <th>Name</th> <th>Key</th> <th>Description</th> <th>Creation Time</th> <th>Last Updated</th> <th>Operation</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>super</td> <td><a href="#">Copy</a> </td> <td></td> <td>2021-12-02 20:10:59</td> <td>2021-12-02 20:10:59</td> <td><a href="#">View Key</a> <a href="#">View Permission</a> <a href="#">Edit</a> <a href="#">Delete</a></td> </tr> </tbody> </table> </div>	<input type="checkbox"/>	Name	Key	Description	Creation Time	Last Updated	Operation	<input type="checkbox"/>	super	<a href="#">Copy</a> 		2021-12-02 20:10:59	2021-12-02 20:10:59	<a href="#">View Key</a> <a href="#">View Permission</a> <a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	Name	Key	Description	Creation Time	Last Updated	Operation									
<input type="checkbox"/>	super	<a href="#">Copy</a> 		2021-12-02 20:10:59	2021-12-02 20:10:59	<a href="#">View Key</a> <a href="#">View Permission</a> <a href="#">Edit</a> <a href="#">Delete</a>									

### 2. Send messages



Messages can be sent in the sync, async, or one-way mode.

- Sync sending

```
for (int i = 0; i < 10; i++) {
    // Create a message instance and set the topic and message content
    Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ " + i).getBytes(
        RemotingHelper.DEFAULT_CHARSET));
    // Send the message
    SendResult sendResult = producer.send(msg);
    System.out.printf("%s%n", sendResult);
}
```

Parameter	Description
topic_name	Topic name, which can be copied under the <b>Topic</b> tab on the cluster details page in the console.
TAG	A parameter used to set the message tag.

- Async sending

```
// Disable retry upon sending failures
producer.setRetryTimesWhenSendAsyncFailed(0);
// Set the number of messages to be sent
int messageCount = 10;
final CountdownLatch countdownLatch = new CountdownLatch(messageCount);
for (int i = 0; i < messageCount; i++) {
    try {
        final int index = i;
        // Create a message instance and set the topic and message content
        Message msg = new Message(topic_name, "TAG", ("Hello rocketMq " + index).getBytes(
            RemotingHelper.DEFAULT_CHARSET));
        producer.send(msg, new SendCallback() {
            @Override
            public void onSuccess(SendResult sendResult) {
                // Logic for message sending successes
                countdownLatch.countDown();
                System.out.printf("%-10d OK %s %n", index, sendResult.getMsgId());
            }

            @Override
            public void onException(Throwable e) {
                // Logic for message sending failures
                countdownLatch.countDown();
                System.out.printf("%-10d Exception %s %n", index, e);
            }
        });
    }
}
```

```
e.printStackTrace();
}
});
} catch (Exception e) {
e.printStackTrace();
}
}
countDownLatch.await(5, TimeUnit.SECONDS);
```

Parameter	Description
topic_name	Topic name, which can be copied under the <b>Topic</b> tab on the cluster details page in the console.
TAG	A parameter used to set the message tag.

- One-way sending

```
for (int i = 0; i < 10; i++) {
// Create a message instance and set the topic and message content
Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ " + i).getBytes
(RemotingHelper.DEFAULT_CHARSET));
Send one-way messages
producer.sendOneway(msg);
}
```

Parameter	Description
topic_name	Topic name, which can be copied under the <b>Topic</b> tab on the cluster details page in the console.
TAG	A parameter used to set the message tag.

Note :

For more information on batch sending or other scenarios, see [Demo](#) or [RocketMQ documentation](#).

## Step 3. Consume messages

### 1. Create a consumer

TDMQ for RocketMQ supports two consumption modes: push and pull.

- For consumers using the push mode:

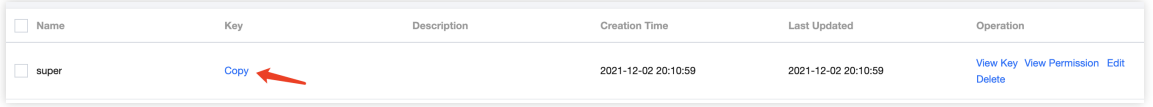
```
// Instantiate the consumer
DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
    namespace,
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))); //ACL per
mission
// Set the NameServer address
pushConsumer.setNamesrvAddr(nameserver);
```

Parameter	Description												
namespace	Namespace name, which can be copied on the <b>Namespace</b> tab on the cluster details page in the console. Its format is <b>cluster ID +   + namespace</b> .												
groupName	Producer group name, which can be copied under the <b>Group</b> tab on the cluster details page in the console.												
nameserver	Cluster access address, which can be copied from <b>Access Address</b> in the <b>Operation</b> column on the <b>Cluster</b> page in the console.												
secretKey	Role name, which can be copied on the <a href="#">Role Management</a> page.												
accessKey	Role token, which can be copied in the <b>Token</b> column on the <a href="#">Role Management</a> page. <div data-bbox="338 1151 1497 1256" data-label="Image"> <table border="1"> <thead> <tr> <th>Name</th> <th>Key</th> <th>Description</th> <th>Creation Time</th> <th>Last Updated</th> <th>Operation</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> super</td> <td>Copy</td> <td></td> <td>2021-12-02 20:10:59</td> <td>2021-12-02 20:10:59</td> <td><a href="#">View Key</a> <a href="#">View Permission</a> <a href="#">Edit</a> <a href="#">Delete</a></td> </tr> </tbody> </table> </div>	Name	Key	Description	Creation Time	Last Updated	Operation	<input type="checkbox"/> super	Copy		2021-12-02 20:10:59	2021-12-02 20:10:59	<a href="#">View Key</a> <a href="#">View Permission</a> <a href="#">Edit</a> <a href="#">Delete</a>
Name	Key	Description	Creation Time	Last Updated	Operation								
<input type="checkbox"/> super	Copy		2021-12-02 20:10:59	2021-12-02 20:10:59	<a href="#">View Key</a> <a href="#">View Permission</a> <a href="#">Edit</a> <a href="#">Delete</a>								

- For consumers using the pull mode:

```
// Instantiate the consumer
DefaultLitePullConsumer pullConsumer = new DefaultLitePullConsumer(
    namespace,
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey, secretKey)));
// Set the NameServer address
pullConsumer.setNamesrvAddr(nameserver);
// Specify the first offset as the start offset for consumption
pullConsumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
```

Parameter	Description
namespace	Namespace name, which can be copied on the <b>Namespace</b> tab on the cluster details page in the console. Its format is <b>cluster ID +   + namespace</b> .

Parameter	Description
groupName	Producer group name, which can be copied under the <b>Group</b> tab on the cluster details page in the console.
nameserver	Cluster access address, which can be copied from <b>Access Address</b> in the <b>Operation</b> column on the <b>Cluster</b> page in the console.
secretKey	Role name, which can be copied on the <a href="#">Role Management</a> page.
accessKey	Role token, which can be copied in the <b>Token</b> column on the <a href="#">Role Management</a> page. 

Note :

For more consumption mode information, see [Demo](#) or [RocketMQ documentation](#).

## 2. Subscribe to messages

The subscription modes vary by consumption mode.

- Subscription in push mode

```
// Subscribe to a topic
pushConsumer.subscribe(topic_name, "*");
// Register a callback implementation class to process messages pulled from the broker
pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
// Message processing logic
System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
// Mark the message as being successfully consumed and return the consumption status
return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
// Start the consumer instance
pushConsumer.start();
```

Parameter	Description
-----------	-------------

Parameter	Description
topic_name	Topic name, which can be copied under the <b>Topic</b> tab on the cluster details page in the console.
"*"	If the subscription expression is left empty or specified as asterisk (*), all messages are subscribed to. `tag1    tag2    tag3` means subscribing to multiple types of tags.

- Subscription in pull mode

```
// Subscribe to a topic
pullConsumer.subscribe(topic_name, "*");
// Start the consumer instance
pullConsumer.start();
try {
    System.out.printf("Consumer Started.%n");
    while (true) {
        // Pull the message
        List<messageext> messageExts = pullConsumer.poll();
        System.out.printf("%s%n", messageExts);
    }
} finally {
    pullConsumer.shutdown();
}
```

Parameter	Description
topic_name	Topic name, which can be copied under the <b>Topic</b> tab on the cluster details page in the console.
"*"	If the subscription expression is left empty or specified as asterisk (*), all messages are subscribed to. `tag1    tag2    tag3` means subscribing to multiple types of tags.

#### Step 4. View consumption details

Log in to the [TDMQ console](#), go to the **Cluster > Group** page, and view the list of clients connected to the group.

Click **View Details** in the **Operation** column to view consumer details.

Basic Info   Namespace   Topic   **Group**

Current Namespace:    Message Retention Period: **1 day**   Max TPS: **8000**

Create (2/10000)   Search by keyword

Group Name	Consumer Info	Consumption Mode	Description	Operation
group-364733	Online Consumer <b>0</b> TPS <b>0</b> Total Heap <b>0</b> <input type="button" value="↻"/>	Unknown		<a href="#">Consumer Details</a> <a href="#">Reset Offset</a> <a href="#">Edit</a> <a href="#">Delete</a>

**Basic Info**

Group Name	group-364733	Creation Time	2022-03-11 15:13:15
Consumption Mode	Unknown	Client Protocol	TCP
Total Heaped Messages	0	Consumer Type	Unknown

**Connected Client**

Client Address	Client Language	Client Version	Message Heap	Operation
No data yet				

Total Items: 0   20 / page     1 / 1 page

Note :

Above is a brief introduction to message publishing and subscription. For more information, see [Demo](#) or [RocketMQ documentation](#)