

Cloud Message Queue

SDK

Product Introduction



Copyright Notice

©2013-2018 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

SDK

[SDK Download](#)

[SDK Instructions](#)

[SDK Update Logs](#)

SDK

SDK Download

Last updated : 2018-09-11 09:35:43

Downloading SDK

Tencent Cloud's message queue supports java, python, php, and C++ SDKs currently, and will support more languages later. You are also welcome to develop more SDK versions of other languages based on API descriptions.

Here are the download links:

- [java sdk](#)
- [python sdk](#)
- [php sdk](#)
- [c++ sdk](#)

Notes about SDK

Before using a SDK, you need to at least get the [secret id](#), [secret key](#), and endpoint (it indicates the region to which the request is to be sent, and whether the request is sent through a private network or a public network).

Instructions for endpoint:

Private network endpoint: `http://cmq-queue-region.api.tencentyun.com`

Public network endpoint: `http(s)://cmq-queue-region.api.qcloud.com`

- If the business process is also deployed on a Tencent Cloud CVM, we strongly recommend that you use a private network endpoint in the same region. For example, if your business process is deployed on a CVM of Tencent Cloud in Beijing, we recommend that you use `http://cmq-queue-bj.api.tencentyun.com`.

Reasons: 1) The latency is lower for a private network in the same region; 2) Currently, message queue

charges a fee for the downstream traffic of public networks, so using a private network can save the cost.

- region should be replaced with a specific region: gz (Guangzhou), sh (Shanghai), or bj (Beijing). The region value in the common parameters should be consistent with the region value of the domain. If there is an inconsistency, the request will be sent to the region specified by the domain.
- Public network domain requests support both HTTP and HTTPS. Private network requests only support HTTP.

SDK Instructions

Last updated : 2018-09-11 09:34:21

Example: Introduction to JAVA SDK (Windows)

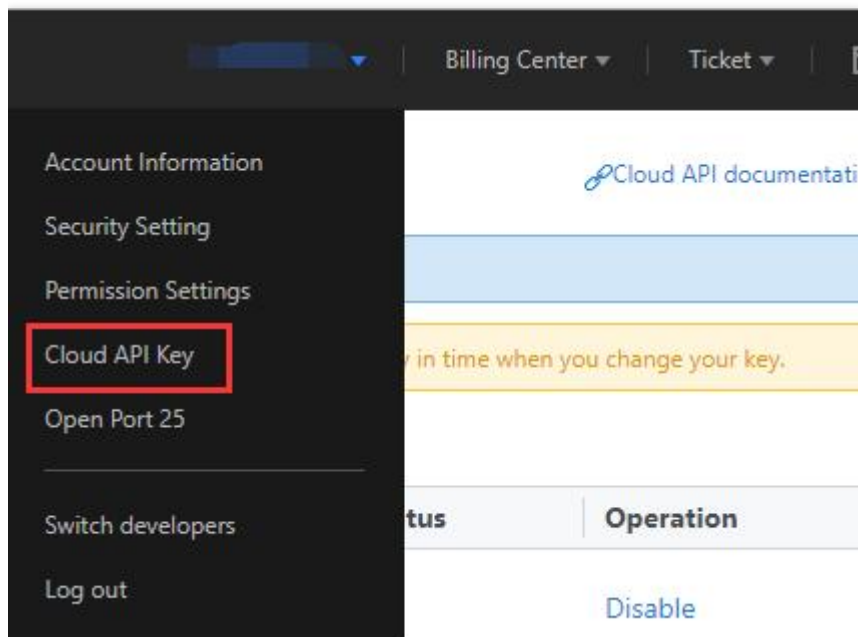
Environment Dependency

Please ensure that JDK environment has been installed. If not, please download [JDK Installer](#) on the Oracle website and install it;

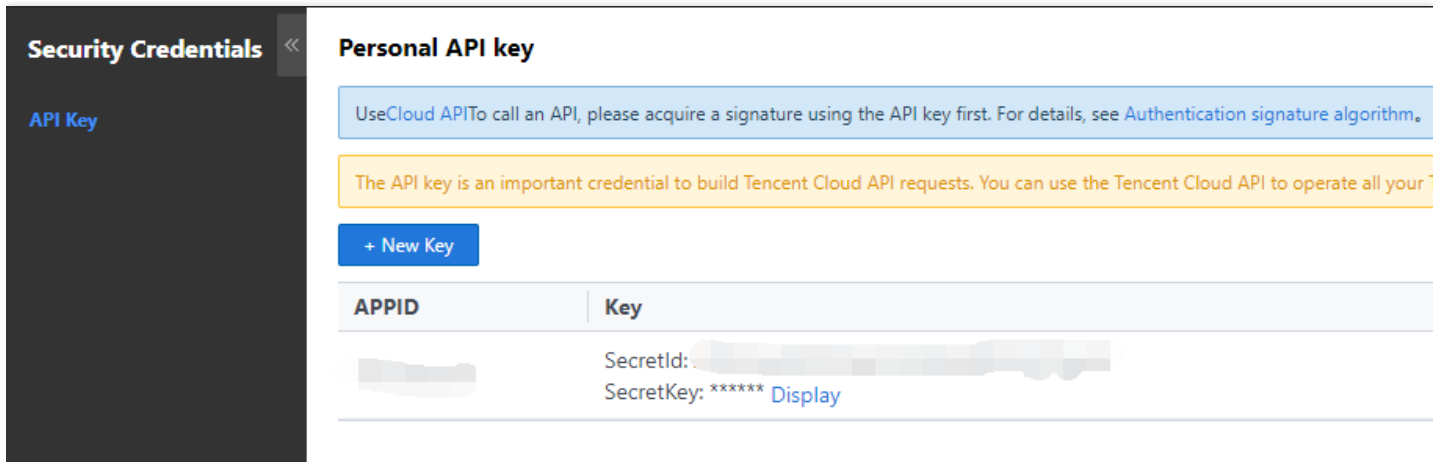
Downloading and Configuring CMQ Java SDK

How to Use Cloud API Key

When using a Java SDK, the user's Cloud API key is required to verify the validity of the user's identity. Users can log in to [Tencent Cloud Console](#), then select account name in the top right corner on the navigation bar, and choose "Cloud API Key" in the drop-down box to access the Cloud API key management page.



Users can create a new Cloud API key or use an existing key. Click key ID and go to the details page to get the secretId of the key and its corresponding secretKey.



endpoint

endpoint is the access address for CMQ service, and it contains the protocol used. endpoint is formatted as follows:

- Private network: `http://cmq-queue-region.api.tencentyun.com`
- Public network: `http(s)://cmq-queue-region.api.qcloud.com`

region

"region" needs to be replaced with a specific region: gz (Guangzhou), sh (Shanghai), bj (Beijing). The option of different regions allows users to choose a region nearby to enjoy better services. "region" value in the common parameters should be consistent with that of domain name. For any inconsistency, the request is sent to the region specified by the domain name.

Differences Between Private and Public Networks

If the business process is also deployed on a Tencent Cloud CVM submachine, we strongly recommend that you use a private network endpoint in the same region:

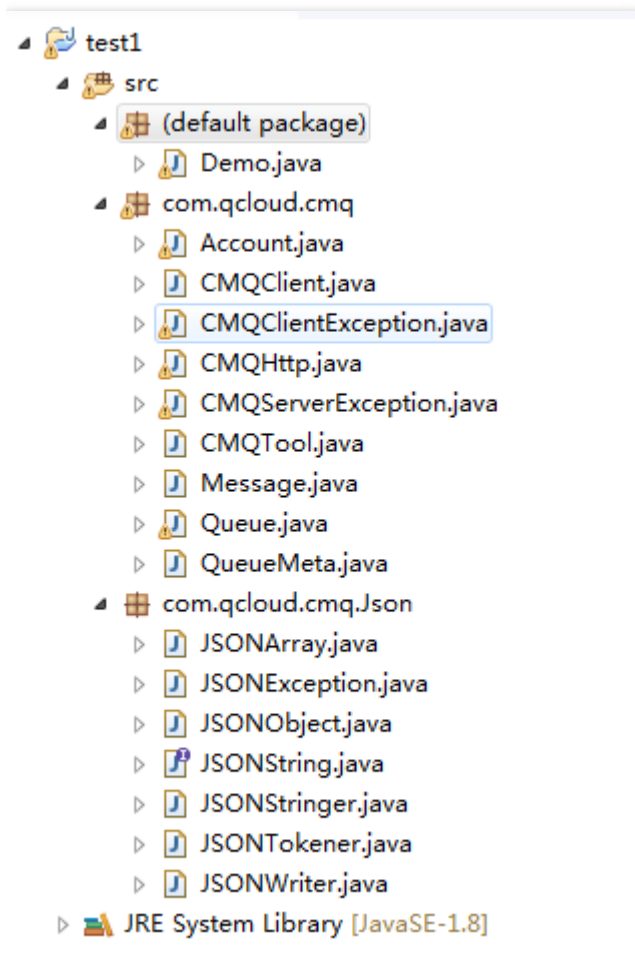
- 1) The latency is lower for a private network in the same region;
- 2) For now, message queue charges a fee for the downstream traffic of public networks, so using a private network can save the cost.

Public network domain requests support both http and https, while private network domain requests only support http. For instance, if your business process is deployed on a CVM of Tencent Cloud in Beijing, we recommend that you request an endpoint in Beijing for shorter latency, and use a private network for low costs. Thus, a proper endpoint you can select is `http://cmq-queue-bj.api.tencentyun.com`.

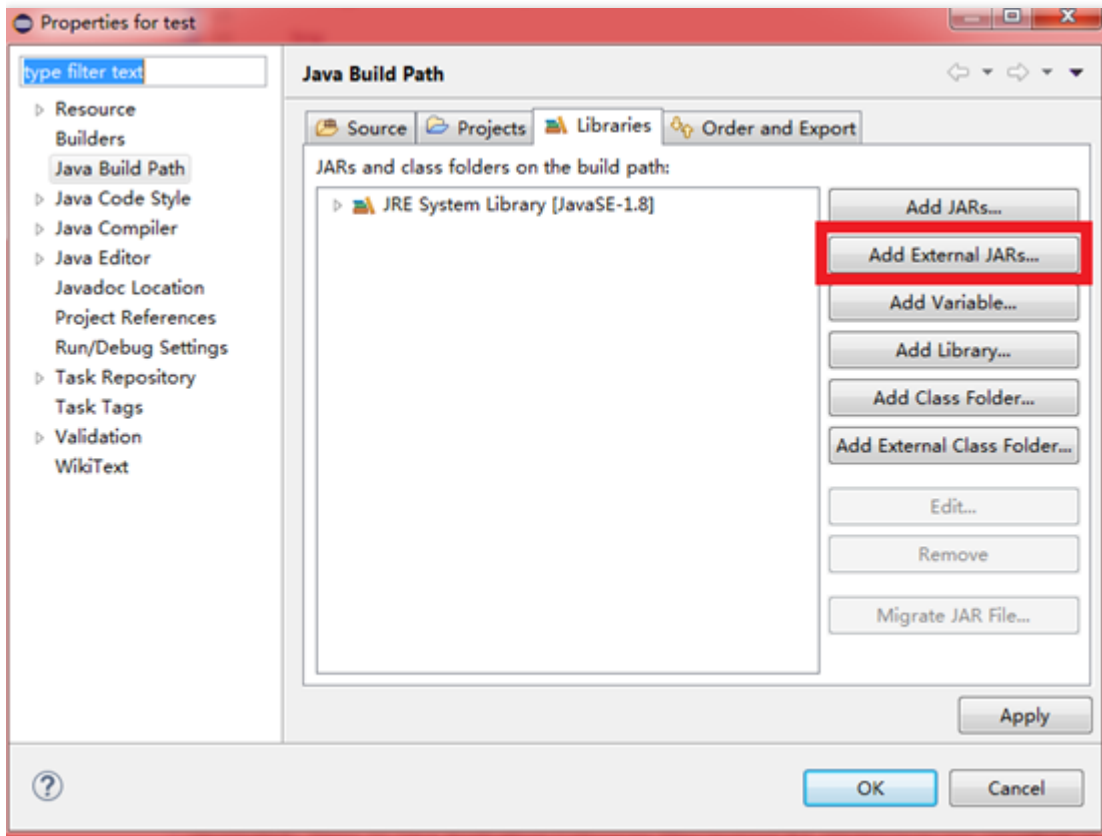
Downloading JAVA SDK

Download the latest version of [CMQ Java SDK](#), or download a [jar Package](#).

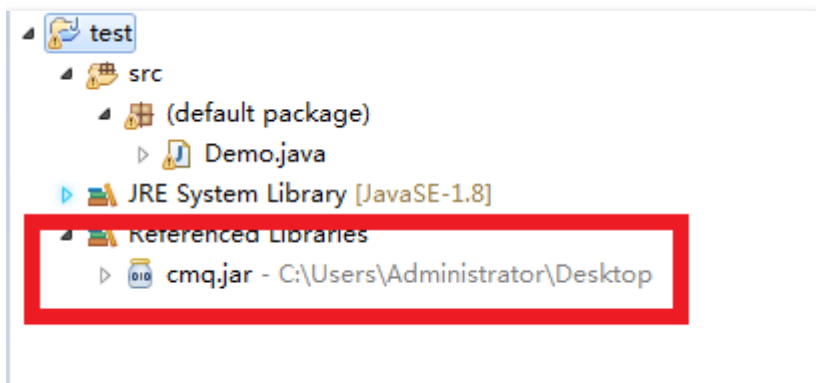
If the java source code is used, directly include the source code in the code directory:



If the jar package is used, please add cmq.jar package through "property" dialog box - "Java Build Path" - "Libraries".



After the jar package is added, the directory will be as follows:



Then you can run the program. If an error code is returned, please refer to the [Error Codes](#) for troubleshooting.

Using CMQ JAVA SDK

The codes below are also samples in Java SDK, demonstrating the whole process of message queue operations including Create a Queue, Get Queue Attributes, Send Messages, Receive Messages, Delete Messages, and Delete a Queue.

```
import com.qcloud.cmq.*;
import java.lang.*;
import java.util.ArrayList;
import java.util.List;
public class
{
public static void main(String[] args) {
String secretId="";
String secretKey="";
String endpoint = "http://cmq-queue-gz.api.qcloud.com";
String path = "/v2/index.php";
String method = "POST";

try
{
Account account = new Account(endpoint,secretId, secretKey);

account.deleteQueue("queue-test10");
System.out.println("-----create queue ...-----");
QueueMeta meta = new QueueMeta();
meta.pollingWaitSeconds = 10;
meta.visibilityTimeout = 10;
meta.maxMsgSize = 65536;
meta.msgRetentionSeconds = 345600;
account.createQueue("queue-test10",meta);
System.out.println("queue-test10 created");
account.createQueue("queue-test11",meta);
System.out.println("queue-test11 created");
account.createQueue("queue-test12",meta);
System.out.println("queue-test12 created");

System.out.println("-----list queue ...-----");
ArrayList<String> vtQueue = new ArrayList<String>();
int totalCount = account.listQueue("",-1,-1,vtQueue);
System.out.println("totalCount:" + totalCount);
for(int i=0;i<vtQueue.size();i++)
{
System.out.println("queueName:" + vtQueue.get(i));
}

System.out.println("-----delete queue ...-----");
account.deleteQueue("queue-test11");
System.out.println("queue-test11 deleted");
account.deleteQueue("queue-test12");
```

```
System.out.println("queue-test12 deleted");

System.out.println("----- queue[queue-test10] -----");
Queue queue = account.getQueue("queue-test10");

System.out.println("-----set queue attributes -----");
QueueMeta meta1 = new QueueMeta();
meta1.pollingWaitSeconds = 20;
queue.setQueueAttributes(meta1);
System.out.println("pollingWaitSeconds=20 set");

System.out.println("-----get queue attributes -----");
QueueMeta meta2 = queue.getQueueAttributes();
System.out.println("maxMsgHeapNum:" + meta2.maxMsgHeapNum);
System.out.println("pollingWaitSeconds:" + meta2.pollingWaitSeconds);
System.out.println("visibilityTimeout:" + meta2.visibilityTimeout);
System.out.println("maxMsgSize:" + meta2.maxMsgSize);
System.out.println("createTime:" + meta2.createTime);
System.out.println("lastModifyTime:" + meta2.lastModifyTime);
System.out.println("activeMsgNum:" + meta2.activeMsgNum);
System.out.println("inactiveMsgNum:" + meta2.inactiveMsgNum);

System.out.println("-----send message -----");
String msgId = queue.sendMessage("hello world,this is cmq sdk for java");
System.out.println("[hello world,this is cmq sdk for java] sent");

System.out.println("-----recv message -----");
Message msg = queue.receiveMessage(10);

System.out.println("msgId:" + msg.msgId);
System.out.println("msgBody:" + msg.msgBody);
System.out.println("receiptHandle:" + msg.receiptHandle);
System.out.println("enqueueTime:" + msg.enqueueTime);
System.out.println("nextVisibleTime:" + msg.nextVisibleTime);
System.out.println("firstDequeueTime:" + msg.firstDequeueTime);
System.out.println("dequeueCount:" + msg.dequeueCount);

System.out.println("-----delete message -----");
queue.deleteMessage(msg.receiptHandle);
System.out.println("receiptHandle:" + msg.receiptHandle + " deleted");

System.out.println("-----batch send message -----");
ArrayList<String> vtMsgBody = new ArrayList<String>();
String msgBody = "hello world,this is cmq sdk for java 1";
vtMsgBody.add(msgBody);
```

```
msgBody = "hello world,this is cmq sdk for java 2";
vtMsgBody.add(msgBody);
msgBody = "hello world,this is cmq sdk for java 3";
vtMsgBody.add(msgBody);
List<String> vtMsgId = queue.batchSendMessage(vtMsgBody);
for(int i=0;i<vtMsgBody.size();i++)
System.out.println "[" + vtMsgBody.get(i) + "] sent");
for(int i=0;i<vtMsgId.size();i++)
System.out.println("msgId:" + vtMsgId.get(i));

ArrayList<String> vtReceiptHandle = new ArrayList<String>();
System.out.println("-----batch rcv message ...-----");
List<Message> msgList = queue.batchReceiveMessage(10,10);
System.out.println("rcv msg count:" + msgList.size());
for(int i=0;i<msgList.size();i++)
{
Message msg1 = msgList.get(i);
System.out.println("msgId:" + msg1.msgId);
System.out.println("msgBody:" + msg1.msgBody);
System.out.println("receiptHandle:" + msg1.receiptHandle);
System.out.println("enqueueTime:" + msg1.enqueueTime);
System.out.println("nextVisibleTime:" + msg1.nextVisibleTime);
System.out.println("firstDequeueTime:" + msg1.firstDequeueTime);
System.out.println("dequeueCount:" + msg1.dequeueCount);

vtReceiptHandle.add(msg1.receiptHandle);
}

queue.batchDeleteMessage(vtReceiptHandle);
System.out.println("-----batch delete message ...-----");
for(int i=0;i<vtReceiptHandle.size();i++)
System.out.println("receiptHandle:" + vtReceiptHandle.get(i) + " deleted");

}
catch(CMQServerException e1){
System.out.println("Server Exception, " + e1.toString());
} catch(CMQClientException e2){
System.out.println("Client Exception, " + e2.toString());
}
}
catch (Exception e) {
System.out.println("error..." + e.toString());
}
}
}
```

SDK Update Logs

Last updated : 2017-12-19 16:08:18

CMQ SDK1.0.3

March 13, 2017

New features for CMQ: message rewind, message delay and routing subscription.

CMQ SDK1.0.2

December 1, 2016

1. Topic and subscription models were supported for SDKs;
2. cmake was used for C++ SDK to manage projects;
3. maven was used for Java SDK to manage projects.

CMQ SDK1.0.1

October 12, 2016

1. Client timeout experience was optimized;
2. The bug that "php sdk authentication failed" was fixed;
3. The bug in php message sending was fixed.

CMQ SDK1.0.0

September 7, 2016

1. Four language versions of c++, java, python, php are released at the same time;
2. SDK encapsulated APIs including Send Messages, Receive Messages and Delete Messages.