消息队列 CMQ

SDK

产品文档

腾讯云

文档目录

SDK
SDK 下载

最近更新时间：2018-09-11 09:35:43

SDK 下载

腾讯云消息队列目前支持 java、python、php 及 C++ SDK，后续会支持更多语言。也欢迎广大开发者根据 API 说明开发更多语言的 SDK 版本。

> **注意：** 由于分配资源和释放资源有 1s 左右的时间，当前消息队列SDK在创建及删除队列/主题时会有 1s 延迟，建议在程序中增加创建和删除的时间间隔保障调用成功。

GitHub 地址如下：

- Java SDK
- Python SDK
- PHP SDK
- C++ SDK

下载地址如下：

- Java SDK
- Python SDK
- PHP SDK
- C++ SDK
- C# SDK

SDK 使用注意事项

使用 SDK 前至少要获取 secret id、 secret key 和 endpoint（即请求发到哪个地域，走内网还是外网）。

endpoint 说明：

队列模型

请参照下面说明将域名中的 {$region} 替换成相应地域：

- 外网接口请求域名： `http(s)://cmq-queue-{$region}.api.qcloud.com`
- 内网接口请求域名： `http://cmq-queue-{$region}.api.tencentyun.com`

主题模型

请参照下面说明将域名中的 {$region} 替换成相应地域：

- 外网接口请求域名： `http(s)://cmq-topic-{$region}.api.qcloud.com`
- 内网接口请求域名： `http://cmq-topic-{$region}.api.tencentyun.com`

如果业务进程也部署在腾讯云的 CVM 子机上，强烈建议使用同地域的内网 endpoint。例如在腾讯云北京地域的 CVM 子机则建议您使用 `http://cmq-queue-bj.api.tencentyun.com` 。原因是：

- 同地域内网时延更低；
- 目前消息队列对于公网下行流量是要收取流量费用的，用内网可以节省这部分的费用。

{$region}需用具体地域替换：gz（广州），sh（上海），bj（北京），shjr（上海金融），szjr（深圳金融），hk（香港）,ca(北美)，cd（成都），usw（美西），sg（新加坡）。公共参数中的 region 值要与域名的 region 值保持一致，如果出现不一致的情况，以域名的 region 值为准，将请求发往域名 region 所指定的地域。

外网域名请求既支持 http，也支持 https。内网请求仅支持 http。

SDK 使用说明

最近更新时间：2018-09-11 09:34:21

CMQ SDK 使用说明

为了方便开发者更好地使用 CMQ 的 SDK，腾讯云提供以下使用说明文档：

- Java (Windows)
- Python (Linux)
- PHP (Linux)
- C++ (Linux)

示例：JAVA SDK 使用简介(Windows)

环境依赖

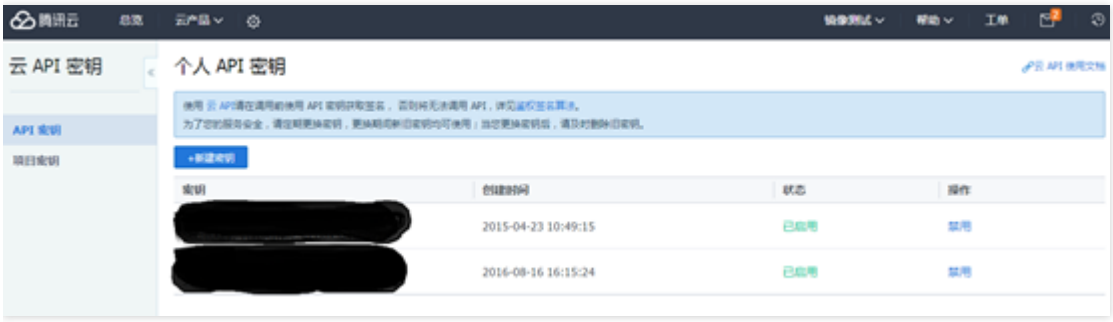
请确保已经安装了 JDK 环境，若未安装请前往 Oracle 官网下载 JDK 安装包 并安装；

CMQ Java SDK 下载与配置

云 API 密钥使用说明

使用 Java SDK 时，首先需要用户的云 API 密钥，云 API 密钥是对用户身份的合法性验证。获取云 API 密钥的方法如下：登录 腾讯云控制台，选择【云产品】>【云 API 密钥】。

用户可在此新建新的云 API 密钥或使用现有密钥。点击密钥 ID 进入详情页获取使用的密钥 secretId 和对应的 secretKey。



endpoint 说明

endpoint 是使用 CMQ 服务的访问地址，同时 endpoint 中也包含了使用的协议，endpoint 的格式如下：

请参照下面说明将域名中的 {$region} 替换成相应地域：

外网接口请求域名：http(s)://cmq-queue-{$region}.api.qcloud.com
内网接口请求域名：http: //cmq-queue-{$region}.api.tencentyun.com

region 说明

{$region}需用具体地域替换：gz（广州），sh（上海），bj（北京），shjr（上海金融），szjr（深圳金融），hk（香港）,ca(北美)，cd（成都），usw（美西），sg（新加坡）。公共参数中的 region 值要与域名的 region 值保持一致，如果出现不一致的情况，以域名的 region 值为准，将请求发往域名 region 所指定的地域。

内外网区别

如果业务进程也部署在腾讯云的 CVM 子机上，强烈建议使用同地域的内网 endpoint，原因如下：

- 同地域内网的时延更低；

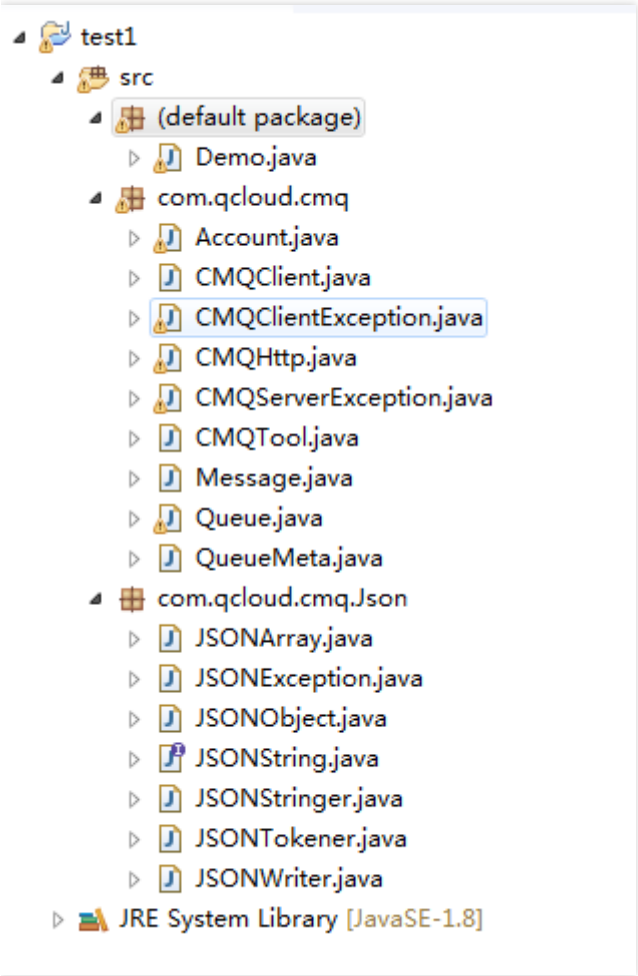
- 目前消息队列对于公网下行流量是要收取流量费用的，用内网可以节省这部分的费用。

外网域名请求既支持 http，也支持 https。内网请求仅支持 http。举个例子：如果使用腾讯云北京地区的云主机，那么建议请求北京地域的 endpoint，这样可以取得较低的时延，同时使用内网可以减少使用费用。因此选用的 endpoint 为 `http://cmq-queue-bj.api.tencentyun.com` 。

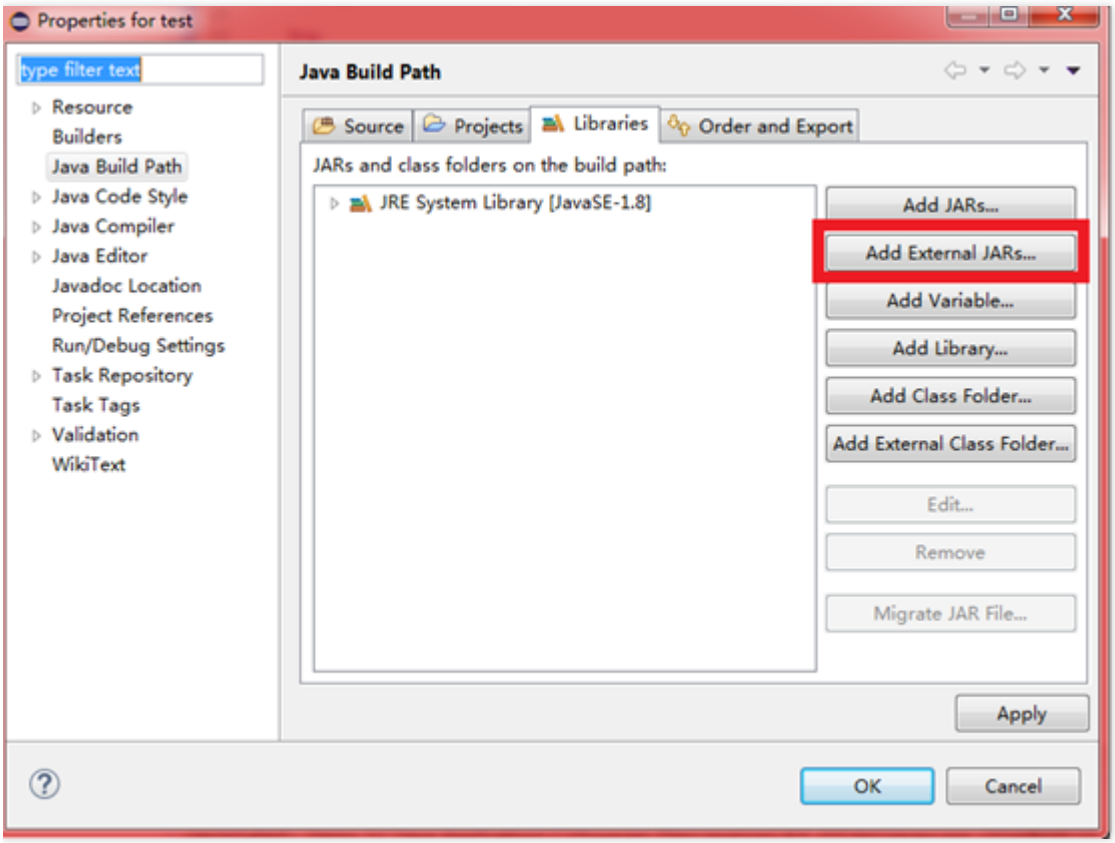
JAVA SDK 下载

下载最新版 CMQ Java SDK，或选择下载 jar 包。

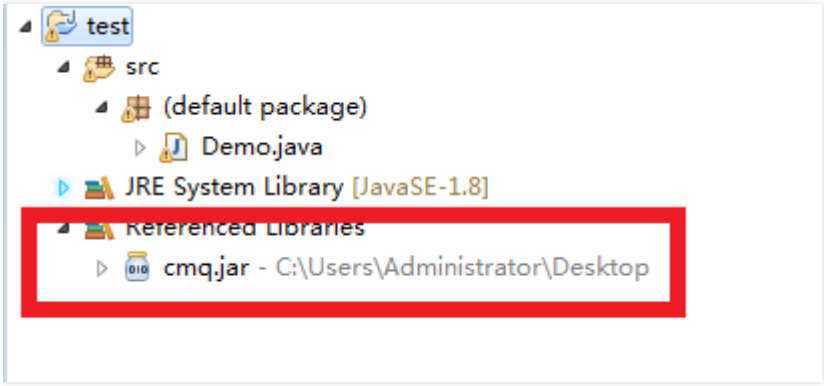
如果使用 java 源码，直接将源码包含在代码目录下：



如果使用 jar 包，请在项目【property】对话框>【Java Build Path】>【Libraries】中加入 cmq.jar 包。



添加 jar 包之后，目录如下：



添加完毕后，就可以运行程序了。如果有错误返回，请参考官网 错误码说明 排查问题。

使用 CMQ JAVA SDK

下面的代码也是 Java SDK 中的sample，从创建队列、获取队列属性、发送消息、接收消息、删除消息、删除队列等操作演示了整个消息队列操作的全过程。

注意： 由于分配资源和释放资源有 1s 左右的时间，当前消息队列 SDK 在创建及删除队列/主题时会有 1s 延迟，建议在程序中增加创建和删除的时间间隔保障调用成功。

```
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 recv message ...---------------");
List<Message> msgList = queue.batchReceiveMessage(10,10);
System.out.println("recv 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 更新日志

最近更新时间：2017-12-19 16:08:18

CMQ SDK1.0.4

2017-4-7

CMQ sdk 支持sha256签名，您可在初始化Account中调用setSignMethod(具体方法名可参考代码说明)设置签名方法,修复已知bug。

CMQ SDK1.0.3

2017-3-13

CMQ新增特性：消息回溯，消息延时，订阅路由功能。

CMQ SDK1.0.2

2016-12-01

1. sdk 支持主题和订阅模式；
2. c++ sdk 改用cmake 管理项目；
3. java sdk 改用 maven 管理项目。

CMQ SDK1.0.1

2016-10-12

1. 优化客户端超时体验；
2. 修复php sdk 鉴权失败bug；
3. 修复php 发送消息bug。

CMQ SDK1.0.0

2016-9-7

1. 同时上线c++，java，python，php四个语言版本；
2. sdk封装了发送消息，接受消息，删除消息等操作接口。