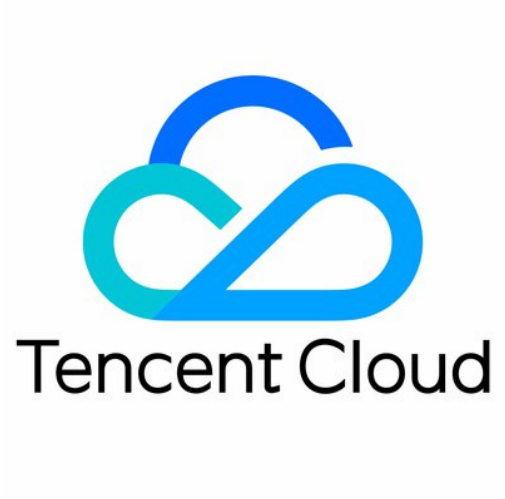
Data Transmission Service

SDK Documentation

Product Documentation



©2013-2019 Tencent Cloud. All rights reserved.

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

Tencent Cloud

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

©2013-2019 Tencent Cloud. All rights reserved.

SDK Documentation
Logs of SDK Releases

Last updated : 2020-06-15 14:25:11

Version 2.9.1

1. Fixed bugs.

Version 2.9.0

1. Supported MySQL 5.7.
2. Supported various encodings including UTF-8.
3. Supported distinction between NULL and empty string.
4. Supported the BLOB field.

Version 2.8.2

1. Optimized SDK memory usage.
2. Supported the VPC authentication, allowing you to use the SDK via both private and public networks.

Version 2.8.0

1. Optimized the internal authentication logic.
2. Reduced the number of user parameters that need to be configured.

Version 2.7.2

1. Filled the `db` value in the DDL statement result.

Version 2.7.0

1. Supported seconds-level HA switch at the data subscription channel backend.
2. Added the SDK monitoring metric report feature.

Version 2.6.0

1. Supported subscribing to multiple channels via a single SDK.
2. Supported subscribing to "stop", "start" and other operations on Client.
3. Supported the serialization of `DataMessage.Record` .
4. Optimized the SDK performance and reduced the resource consumption.

©2013-2019 Tencent Cloud. All rights reserved.

Version 2.5.0

1. Fixed bugs occurred with small probability in high-concurrency scenarios.
2. Supported the globally unique auto-increment ID recorded in transactions.

Version 2.4.0

1. Optimized the subscription logic by working with the backend to accurately display SDK's current consumption time point.
2. Fixed the problem occurred while encoding a few special characters at the backend.
3. Fixed multiple compatibility issues. We recommend that users who use earlier versions upgrade the SDK to this version as soon as possible.

©2013-2019 Tencent Cloud. All rights reserved.

Data Subscription SDK

Last updated : 2020-12-16 14:42:56

Downloading the Data Subscription SDK

Click here to download the data subscription SDK v2.9.1.

How It Works

Fetching messages

The SDK runs two parallel asynchronous threads independent of each other to fetch and acknowledge messages. Messages are fetched in order.

The SDK calls the user-registered `notify` function to notify the fetched messages one by one. Every time the SDK fetches a message, it notifies the message (just once). If the `m.ackAsConsumed()` function is not called, messages are still notified because the fetching thread and acknowledgement thread are asynchronous.

Acknowledgement mechanism

The SDK adopts an incremental acknowledgment mechanism that allows repeat acknowledgments but does not allow missing any message including `BEGIN` and `COMMIT` messages.

- For example, if the client receives message 1, 2, 3, 4, and 5 but it calls `m.ackAsConsumed()` only for message 1, 2, and 5, the SDK will confirm to the server that only message 1 and 2 have been consumed. If the client fails at this point, the SDK will fetch starting from message 3.
  Since the fetching thread and acknowledgement thread are asynchronous, the SDK will keep fetching new messages and notifying them to the client even if some messages are not acknowledged. However, if the number of unacknowledged messages exceeds a threshold (currently 8,000), the SDK will no longer fetch new messages.
- Every message has a unique `record_id` and `checkpoint` , and the SDK acknowledge a message by its `checkpoint` .

Runtime Environment Requirements

- Java: JRE v1.6 or later

©2013-2019 Tencent Cloud. All rights reserved.

- The SDK needs to run on a Tencent Cloud CVM instance in the same VPC and the same region as the subscription instance (if they are not in the same VPC, you need to configure them so that they can connect to each other).
- To access the SDK over a public network, the CVM can be used for port forwarding, but **the bandwidth and performance depend heavily on the public network bandwidth**.

Sample Code

The sample code of Tencent Cloud binlog subscription is as follows:

```
package com.qcloud.biz;
import com.qcloud.dts.context.NetworkEnv;
import com.qcloud.dts.context.SubscribeContext;
import com.qcloud.dts.message.ClusterMessage;
import com.qcloud.dts.message.DataMessage;
import com.qcloud.dts.subscribe.ClusterListener;
import com.qcloud.dts.subscribe.DefaultSubscribeClient;
import com.qcloud.dts.subscribe.SubscribeClient;
import java.util.List;
public class Main {
public static void main(String[] args) throws Exception {
// Create a context
SubscribeContext context=new SubscribeContext();
// User "secretId" and "secretKey"
context.setSecretId("AKID-522dabxxxxxxxxxxxxxxxxx");
context.setSecretKey("AKEY-0ff4cxxxxxxxxxxxxxxxxxx");
// Specify the channel region. The "Region" parameter is required in SDKs later than v2.8.0.
// For valid values of "Region", please see [Common Parameters](https://cloud.tencent.com/documen
t/product/236/15833#.E5.9C.B0.E5.9F.9F.E5.88.97.E8.A1.A8).
context.setRegion("ap-chongqing");
// "serviceIp" and "servicePort" subscribed
// Note that the "Ip" and "Port" parameters are required in SDKs earlier than v2.8.0, but they ca
n be ignored in SDKs later than v2.8.0 if the "Region" parameter is specified.
// context.setServiceIp("10.108.112.24");
// context.setServicePort(50120);

// If the CVM instance where the SDK runs cannot be accessed over a public network, set the "Netw
orkEnv" parameter to private network. The default parameter value is public network.
context.setNetworkEnv(NetworkEnv.LAN);
// Create a client
SubscribeClient client=new DefaultSubscribeClient(context);
// Create a subscription listener
ClusterListener listener= new ClusterListener() {
@Override
```

©2013-2019 Tencent Cloud. All rights reserved.

```
public void notify(List<ClusterMessage> messages) throws Exception {
// Consume subscribed data
for(ClusterMessage m:messages){
for(DataMessage.Record.Field f:m.getRecord().getFieldList()){

if (f.getType().equals(DataMessage.Record.Field.Type.BLOB)){
System.out.println("["+f.getType()+"]["+f.getFieldname()+"] the original value:");
byte[] theRawBytesValue = f.getValueAsBytes();
}if(f.getType().equals(DataMessage.Record.Field.Type.INT8)){
// If the value is null, "f.getValueAsInteger()" returns null.
System.out.println(f.getValueAsInteger());
}if(f.getType().equals(DataMessage.Record.Field.Type.JSON)){
// JSON data can be returned only when the source instance engine is MySQL v5.7.
System.out.println(f.getValueAsString());
}if(f.getType().equals(DataMessage.Record.Field.Type.STRING)){
// If the value is null, "f.getValueAsString()" returns null.
System.out.println(f.getValueAsString());
// The original encoding of the field
System.out.println(f.getFieldEnc());
}
else{
// The "f.getValue()" method will be unsupported soon.
String value = f.getValue() == null ? "Null": f.getValue();
String msg = "["+f.getType()+"]"+f.getFieldname()+"[encoding:"+f.getFieldEnc()+"]"+"[value:"+valu
e+"]";
System.out.println(msg);
}

}
// Acknowledge consumption
m.ackAsConsumed();
}
}
@Override
public void onException(Exception e){
System.out.println("listen exception"+e);
}};
// Add a listener
client.addClusterListener(listener);
// Configure the requested subscription channel
client.askForGUID("dts-channel-r0M8kKsSyRZmSxQt");
// Launch the client
client.start();
}
}
```

©2013-2019 Tencent Cloud. All rights reserved.

The whole process is an intuitive, typical producer-consumer model. As a consumer, the SDK constantly fetches subscribed binlog data from the server, consumes the data, and acknowledges data consumption.

1. Configure parameters and create a client `SubscribeClient` (the consumer).
2. Create a listener `ClusterListener` to consume the received binlog subscription data, and return an acknowledgement after consumption.
3. Launch the client to start the process.
   The listener `ClusterListener` allows you to operate on the received binlog data based on your own needs and filter the data by type, for example, filtering out all `drop` statements.

In the sample code, you need to provide five parameters.

- `secretId` and `secretKey` are the values of keys associated with your Tencent cloud account, which can be viewed in **Access Key** > **API Key** in the CAM console. The SDK uses these two parameters to authenticate your operations.

  > ⚠ **Note :**
  >
  > Data subscription SDK has been connected to CAM. As the root account has all the permissions by default, you can use the API key of the root account to access the SDK. Sub-accounts have no permissions by default, which must be given the access to the operation `name/dts:AuthenticateSubscribeSDK`, or the access to all DTS operations `QcloudDTSFullAccess` by the root account.

- `serviceIp`, `servicePort`, and `channelId` are related to binlog subscription. After you configure subscription in the TencentDB for MySQL console, you can view details in **Data Subscription** in the DTS console.

  > ⓘ **Note :**
  >
  > `serviceIp` is the **Service IP**, `servicePort` the **Service Port**, and `channelId` the **Channel ID** on the subscription details page in **Data Subscription** in the DTS console.

SDK Description

SubscribeContext class

Class description

©2013-2019 Tencent Cloud. All rights reserved.

This is mainly used to set your SDK's configuration information, including the security credential (`secretId` and `secretKey`), and the IP and port of subscription service.

Construction method

public SubscribeContext()

Class method

Setting the security credential (secretId)

Function prototype

public void setSecretId(String secretId)

Input parameters

Parameter Name	Type	Description
secretId	String	Security credential, which can be viewed in **Access Key** > **API Key** in the CAM console.

Response

N/A

Exceptions thrown

N/A

Setting the security credential (secretKey)

Function prototype

public void setSecretKey(String secretKey)

Input parameters

Parameter Name	Type	Description
secretKey	String	Security credential, which can be viewed in **Access Key** > **API Key** in the CAM console

Response

N/A

Exceptions thrown

N/A

©2013-2019 Tencent Cloud. All rights reserved.

Setting the subscription service IP

Function prototype

public void setServiceIp(String serviceIp)

Input parameters

Parameter Name	Type	Description
serviceIp	String	The IP address of subscription service, which can be viewed on the subscription details page in **Data Subscription** in the DTS console

Response

N/A

Exceptions thrown

N/A

Setting the subscription service port

Function prototype

public void setServicePort(String servicePort)

Input parameters

Parameter Name	Type	Description
servicePort	String	The port number of subscription service, which can be viewed on the subscription details page in **Data Subscription** in the DTS console

Response

N/A

Exceptions thrown

N/A

SubscribeClient API and DefaultSubscribeClient API

Class description

The `DefaultSubscribeClient` class implements the `SubscribeClient` API.
This is used to build the client program for the data subscription SDK, i.e. the consumer for binlog messages.

©2013-2019 Tencent Cloud. All rights reserved.

Based on user requirements, `DefaultSubscribeClient` provides two implementation methods, synchronous acknowledgment and asynchronous acknowledgment. In synchronous mode, an acknowledgment is synchronously returned each time the client consumes a binlog message, to ensure that message consumption acknowledgments can be received by the server as soon as possible. In this mode, the overall performance of SDK is lower compared to asynchronous mode. In asynchronous mode, the client acknowledges message consumption asynchronously, that is, messages are fetched and acknowledged asynchronously and independently, in which case performance is higher than that in synchronous mode. You may select either mode as desired.

Construction method

Constructing DefaultSubscribeClient

Function prototype

public DefaultSubscribeClient(SubscribeContext context, boolean isSync) throws Exception

Input parameters

Parameter Name	Type	Description
context	SubscribeContext	Configurations of your SDK
isSynce	boolean	Whether synchronous mode is used for SDK

Response

`DefaultSubscribeClient` instance

Exceptions thrown

- IllegalArgumentException: this exception is thrown when any parameter is invalid in the `context` parameter submitted by a user. Invalid situations: no security credential or incorrect format; no service IP/port or incorrect format.
- Exception: this exception is thrown when an internal error occurred while initializing the SDK.

Constructing DefaultSubscribeClient

Function prototype

public DefaultSubscribeClient(SubscribeContext context) throws Exception

Input parameters

Parameter Name	Type	Description
context	SubscribeContext	Configurations of your SDK

©2013-2019 Tencent Cloud. All rights reserved.

Response

`DefaultSubscribeClient` instance. The default mode is asynchronous mode.

Exceptions thrown

- IllegalArgumentException: this exception is thrown when any parameter is invalid in the `context` parameter submitted by a user. Invalid situations: no security credential or incorrect format; no service IP/port or incorrect format.
- Exception: this exception is thrown when an internal error occurred while initializing the SDK.

Class method

Adding a listener for the SDK client (consumer)

Function description

Add the `ClusterListener` listener to a `SubscribeClient` to subscribe to the incremental data in the channel.

Function prototype

public void addClusterListener(ClusterListener listener) throws Exception

Input parameters

Parameter Name	Type	Description
listener	ClusterListener	Listener used by a consumer client. The main process to consume binlog messages should be implemented in `ClusterListener` .

Response

N/A

Exceptions thrown

- IllegalArgumentException: this exception is thrown when the `listener` parameter submitted by user is empty.
- Exception: the SDK only supports one listener. This exception is thrown when multiple listeners are added.

Requesting incremental data in a subscription channel

Function prototype

public void askForGUID(String channelId)

Input parameters

©2013-2019 Tencent Cloud. All rights reserved.

Parameter Name	Type	Description
channelId	String	Subscription channel ID, which can be viewed on the subscription details page in **Data Subscription** in the DTS console

Response

N/A

Exceptions thrown

N/A

Launching the SDK client

Function prototype

public void start() throws Exception

Input parameters

N/A

Response

N/A

Exceptions thrown

Exception: this exception is thrown if an internal error occurred while launching the SDK.

Stopping the SDK client

Function prototype

public void stop(int waitSeconds) throws Exception

public void stop() throws Exception

Input parameters

Parameter Name	Type	Description
waitSeconds	int	Waiting time (in seconds), which indicates how long it takes to forcedly stop the SDK

The `stop` function with no parameters will wait for a period of time for the thread to stop, which may take longer and should be subject to the system scheduling. We recommend that you use the `stop` function with timeout parameters for scenarios where specific restart time is required.

©2013-2019 Tencent Cloud. All rights reserved.

Response

N/A

Exceptions thrown

Exception: this exception is thrown if an internal error occurred while stopping the SDK.

ClusterListener API

API description

This is a callback API. An SDK user should implement the `notify` function of this API to consume subscription data, and implement the `onException` function to handle exceptions that may occur during the consumption process.

API functions

Notifying the SDK consumer client of subscription messages

Function description

This is mainly used to implement the consumption of incremental data. However, the SDK will notify `ClusterListener` of the subscription data via the `notify` function when the data is received.

Function prototype

public abstract void notify(List<ClusterMessage> messages) throws Exception

Input parameters

Parameter Name	Type	Description
messages	List<ClusterMessage>	Subscription data array. For more information on implementation of `ClusterMessage`, please see its definition.

Response

N/A

Exceptions thrown

Any exception during the consumption of subscription data will be thrown to the `onException` function implemented by users who will then handle these exceptions as needed.

Handling exceptions occurred while consuming subscription data

Function description

This is mainly used to handle exceptions occurred while consuming subscription data. Users can implement their own secure exit policy in `onException`.

©2013-2019 Tencent Cloud. All rights reserved.

Function prototype

public abstract void onException(Exception exception)

Input parameters

Parameter Name	Type	Description
exception	Exception	`Exception` class in Java standard library

Response

N/A

Exceptions thrown

N/A

ClusterMessage class

Class description

The `ClusterMessage` class delivers consumed subscription data through the `notify` function. Each `ClusterMecentssage` saves data records of one **transaction** in TencentDB for MySQL, and each record in the transaction is saved via `Record` .

Class method

Obtaining records from ClusterMessage

Function prototype

public Record getRecord()

Input parameters

N/A

Response

Type	Parameter Description
Record	Change record corresponding to a specific record in a transaction, such as `begin` , `commit` , `update` , `insert` , etc.

Exceptions thrown

N/A

Acknowledging consumed data

Function description

©2013-2019 Tencent Cloud. All rights reserved.

This is used to send an acknowledgment to the subscription server about the consumed data. This function acknowledges synchronously or asynchronously according to the value configured in `SubscribeClient`. Users should always call this function after consumption process, otherwise the SDK may receive duplicate data due to incorrect logic.

> ⚠ **Note :**
>
> The SDK must call `ackAsConsumed` to acknowledge every received message including those that the business logic may not care about, or else the SDK will stop fetching new messages after a certain number of messages remain unacknowledged.

Function prototype

public void ackAsConsumed() throws Exception

Input parameters

N/A

Response

N/A

Exceptions thrown

Exception: this exception is thrown if an internal error occurred during the acknowledgement process.

Record class

Class description

This indicates a certain record in subscribed binlog data, generally, a member of a certain transaction `ClusterMessage`. The record may be a `begin`, `commit` or `update` statement.

Class method

Obtaining the attribute value of Record

Function prototype

public String getAttribute(String key)

Input parameters

Parameter Name	Type	Description
key	String	Name of attribute value

Possible attribute key values are:

©2013-2019 Tencent Cloud. All rights reserved.

Attribute Key Value	Description
record_id	Record ID, which is an auto-increment but non-continuous string in a channel
source_type	Engine type of the database instance of `Record` . Available value: mysql
source_category	Record type. Available value: full_recorded
timestamp	The time when the `Record` is stored into binlog. This is also the time when the SQL statement is executed in TencentDB.
sdkInfo	The check point in the binlog file of corresponding `Record` , in the format of `file_offset@file_name` , where `file_name` is the number suffix of the binlog file
record_type	Operation type of `Record` . Available values: insert/update/delete/replace/ddl/begin/commit/heartbeat
db	Name of the database where the `Record` update table is created. For DDL records, the field is empty.
table_name	Name of `Record` update table. For DDL records, the field is empty.
record_encoding	Encoding of `Record`
primary	Name of the primary key column of `Record` update table
fields_enc	Encoding of each field value of `Record` . Fields are separated by commas, and an empty value is used for non-character type.
gtid	GTID of the transaction of `Record`

Response

Type	Parameter Description
String	Attribute value

Exceptions thrown

N/A

Obtaining the change type of a record

©2013-2019 Tencent Cloud. All rights reserved.

Function prototype

public DataMessage.Record.Type getOpt()

Input parameters

N/A

Response

Type	Parameter Description
DataMessage.Record.Type	Record type. Valid values: insert, delete, update, replace, ddl, begin, commit, heartbeat. `heartbeat` refers to the heartbeat table defined by DTS, which is used to check whether the subscription channel is health. Theoretically, one heartbeat can be generated per second.

Exceptions thrown

N/A

Obtaining the checkpoint of a record in binlog

Function prototype

public String getCheckpoint()

Input parameters

N/A

Response

Type	Parameter Description
String	Checkpoint of a record in binlog, in the format of `binlog_offset@binlog_fid`. `binlog_offset` is the offset of the change record in the binlog file, and `binlog_fid` is the name of the binlog file.

Exceptions thrown

N/A

Obtaining the timestamp of a record in binlog

Function prototype

public String getTimestamp()

Input parameters

N/A

©2013-2019 Tencent Cloud. All rights reserved.

Response

Type	Parameter Description
String	Timestamp string

Exceptions thrown

N/A

Obtaining the database name of a record

Function prototype

public String getDbname()

Input parameters

N/A

Response

Type	Parameter Description
String	Database name string

Exceptions thrown

N/A

Obtaining the data table name of a record

Function prototype

public String getTableName()

Input parameters

N/A

Response

Type	Parameter Description
String	Data table name string

Exceptions thrown

N/A

Obtaining the primary key column name of a record

©2013-2019 Tencent Cloud. All rights reserved.

Function prototype

public String getPrimaryKeys()

Input parameters

N/A

Response

Type	Parameter Description
String	Primary key column name. For composite primary keys, the column names are separated by semicolons.

Exceptions thrown

N/A

Obtaining the database type of a subscription instance

Function prototype

public DBType getDbType()

Input parameters

N/A

Response

Type	Parameter Description
DBType	Only TencentDB for MySQL is supported for data transfer, that is, `DBType.MYSQL` .

Exceptions thrown

N/A

Obtaining the number of fields in Record

Function prototype

public int getFieldCount()

Input parameters

N/A

Response

Type	Parameter Description
int	The number of fields in `Record` , which is equal to the number of columns in the table or

©2013-2019 Tencent Cloud. All rights reserved.

	the double of that (for update `Record`)

Exceptions thrown

N/A

Checking if Record is the first one in a transaction

Function prototype

public Boolean isFirstInLogevent()

Input parameters

N/A

Response

Type	Parameter Description
Boolean	True: it is the first log in the transaction. False: it is not the first log.

Exceptions thrown

N/A

Obtaining the field definition list of a record table

Function prototype

public List<Field> getFieldList()

Input parameters

N/A

Response

Type	Parameter Description
List<Field>	Field array. For more information, please see the definition of the `Field` class.

⚠ Note :

- For `INSERT` records, the `Field` in `List` corresponds to these records in the order defined by the subscription table, and the values of records in `Field` are inserted data (before images).
- For `DELETE` records, the `Field` in `List` corresponds to these records in the order defined by the subscription table, and the values of records in `Field` are deleted data (after images).

©2013-2019 Tencent Cloud. All rights reserved.

- For `UPDATE` records, `List` contains data before and after change (before images and after images). Before images (data before change) are in even positions of `List`, while after images (data after change) in odd positions. The before image list and after image list correspond to records in the order defined by the subscription table, so the number of `Field` in the `List` is twice as many as the number of columns in the subscription table.

Exceptions thrown

N/A

Field class

Class description

The `Field` class defines the attributes of a field such as encoding, type, name, value, and whether it is a primary key.

Class method

Obtaining the encoding format of a field

Function prototype

public String getFieldEnc()

Input parameters

N/A

Response

Type	Parameter Description
String	Field encoding of string type

Exceptions thrown

N/A

Obtaining the field name

Function prototype

public String getFieldname()

Input parameters

N/A

Response

Type	Parameter Description

©2013-2019 Tencent Cloud. All rights reserved.

String	Field name of string type

Exceptions thrown

N/A

Obtaining the data type of a field

Function prototype

public Field.Type getType()

Input parameters

N/A

Response

Type	Parameter Description
Field.Type	Field.Type is an enumeration type which corresponds to data types supported by MySQL, including INT8, INT16, INT24, INT32, INT64, DECIMAL, FLOAT, DOUBLE, NULL, TIMESTAMP, DATE, TIME, DATETIME, YEAR, BIT, ENUM, SET, BLOB, GEOMETRY, STRING, UNKOWN

Exceptions thrown

N/A

Obtaining the field value

Function prototype

public ByteString getFieldname()

Input parameters

N/A

Response

Type	Parameter Description
ByteString	Field value. NULL if the value is empty.

Exceptions thrown

N/A

Checking if the field is a primary key

©2013-2019 Tencent Cloud. All rights reserved.

Function prototype

public Boolean isPrimary()

Input parameters

N/A

Response

Type	Parameter Description
Boolean	True: the field is a primary key. False: it is not a primary key.

Exceptions thrown

N/A

©2013-2019 Tencent Cloud. All rights reserved.

SDK Upgrade

Last updated : 2020-06-15 14:25:11

The API 2.0 will not be available for the data subscription SDK of Tencent Cloud Data Transmission Service (DTS). Data subscription SDK 2.8.0 and earlier versions use the API 2.0 for authentication, and thus will be affected when the API 2.0 is deprecated.

Please follow the steps below to upgrade the applications you have consumed through the SDK in time to avoid the impact of API 2.0 deprecation.

Step 1. Check the SDK version

Check the version of the used SDK you have downloaded from Data Subscription SDK.
Check the SDK version as follows:

- Check the package name of the SDK you have used, which contains the version number, such as `binlogsdk-2.8.0-jar-with-dependencies.jar` .
- Check the value of `SDK_VERSION` . If the SDK package has been renamed, run the `unzip` command to decompress the package and check the value of `SDK_VERSION` in the `tencentSubscribe.properties` file.

If the version number of your data subscription SDK is 2.8.0 or smaller, go to Step 2.

Step 2. Upgrade the SDK

1. Download the latest SDK from Data Subscription SDK and replace SDK 2.8.0 and earlier versions with it.
2. After the replacement, refer to the following "Code modification starts" to add a line of code to set `region` for subscription channel.

```
package com.qcloud.biz;
import com.qcloud.dts.context.NetworkEnv;
import com.qcloud.dts.context.SubscribeContext;
import com.qcloud.dts.message.ClusterMessage;
import com.qcloud.dts.message.DataMessage;
import com.qcloud.dts.subscribe.ClusterListener;
import com.qcloud.dts.subscribe.DefaultSubscribeClient;
import com.qcloud.dts.subscribe.SubscribeClient;
```

©2013-2019 Tencent Cloud. All rights reserved.

```
import java.util.List;
public class Main {
public static void main(String[] args) throws Exception {

SubscribeContext context=new SubscribeContext();

context.setSecretId("AKID-522dabxxxxxxxxxxxxxxxx");
context.setSecretKey("AKEY-0ff4cxxxxxxxxxxxxxxxxx");
/*****************Code modification starts****************/
// If you do not set the `region` parameter, the API 2.0 will still be used for authentication ev
en after it is deprecated.
// However, the authentication via API 2.0 will fail then, so the SDK will not work properly.
// Set `region` for subscription channel.
context.setRegion("ap-chongqing");
/*****************Code modification ends*****************/


// Create a client
SubscribeClient client=new DefaultSubscribeClient(context);
// Create a subscription listener
ClusterListener listener= new ClusterListener() {
@Override
public void notify(List<ClusterMessage> messages) throws Exception {
// Consume subscribed data
for(ClusterMessage m:messages){
for(DataMessage.Record.Field f:m.getRecord().getFieldList()){
if(f.getFieldname().equals("id")){
System.out.println("seq:"+f.getValue());
}
DataMessage.Record record = m.getRecord();
}
// Acknowledge consumption
m.ackAsConsumed();
}
}
@Override
public void onException(Exception e){
System.out.println("listen exception"+e);
}};
// Add a listener
client.addClusterListener(listener);
// Configure requested subscription channel
client.askForGUID("dts-channel-r0M8kKsSyRZmSxQt");
// Start the client
client.start();
}
}
```

©2013-2019 Tencent Cloud. All rights reserved.