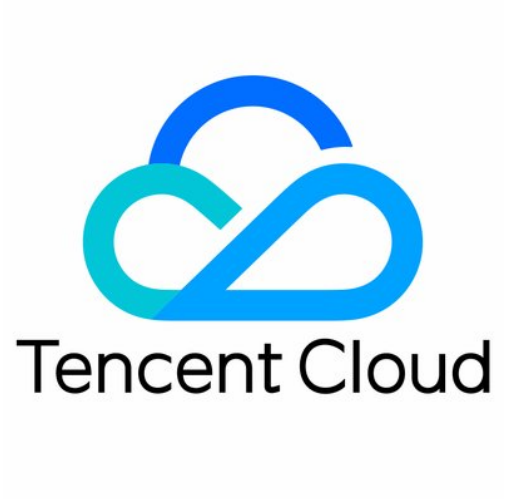
TencentDB for TcaplusDB

TcaplusDB SDK

Product Documentation



©2013-2022 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2024 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

TcaplusDB SDK
- Release History
- SDK Download
- C++ SDK API
- TcaplusDB Error Codes
- SDK Installation
  - Installation Guide for Linux
  - Installation Guide for Windows
- Directions for Protobuf Table SDK for C++
  - Writing Data
  - Reading Data
  - Updating Data
  - Deleting Data
- Directions for TDR Table SDK for C++
  - Writing Data
  - Reading Data
  - Updating Data
  - Deleting Data

©2013-2022 Tencent Cloud. All rights reserved.

TcaplusDB SDK Release History

Last updated：2020-04-02 09:52:31

TcaplusDB SDK 3.34.0.191456

Release date: April 21, 2019

New:

- pb index tables can be created, and batch query can be performed on indices.
- `callback` objects can be released for optimization in a self-service manner.
- The async mode of packet sending/receipt is optimized.
- The log feature is optimized.

Fixes:

- The `init` failure of multithreaded API objects is fixed.
- The error where GET requests couldn't support the default GET version is fixed.
- The error where repeated reconnections were made after disconnection between API and `dir` is fixed.
- The error where large `dir` return packets couldn't be called back is fixed.

TcaplusDB SDK 3.32.0.185732

Release date: November 22, 2018

Fixes:

- The multithreaded traversal issue is fixed.

TcaplusDB SDK 3.32.0.174230

Release date: June 6, 2018

Fixes:

- The `RegistAllTable` error is fixed.

©2013-2022 Tencent Cloud. All rights reserved.

SDK Download

Last updated：2023-06-13 16:57:54

Overview

This document describes how to download TcaplusDB SDK.

Updates

Before downloading the SDK, see the Release History.

©2013-2022 Tencent Cloud. All rights reserved.

C++ SDK API

Last updated：2020-07-15 11:27:53

Overview

TcaplusDB service API is the data access entry for your application to access TcaplusDB and the programming API for your application to access business data in TcaplusDB. Currently, TcaplusDB mainly uses Google Protocol Buffer (Protobuf) to implement communication and data element definition.

Process

After you activate the service in the console and create a table, the details page will display the access ID, access password, and private IP address, and the Protobuf table management page will provide information such as the name of the created table and table group ID. You can use TcaplusDB API for C++ to enable your application to manipulate multiple tables in the cluster.

Module

You can use the Protobuf protocol to define tables compliant with the TcaplusDB specifications. You can create a table or modify an existing table by transferring the metafile of the table definition to the TcaplusDB Console. After the operation succeeded, you can use TcaplusDB Protobuf API to read/write table data records. Currently, TcaplusDB Protobuf API supports the following operations:

Operation	Feature Description
GET	Gets the information of one value based on one field
BATCHGET	Gets the information of multiple values based on multiple fields
ADD	Inserts a data entry. If the record of the corresponding field already exists, an error will be returned
SET	Updates data if the record corresponding to the field exists or inserts data request if the record does not exist
DEL	Deletes corresponding record based on field
FIELDINC	Adds specified value based on specified field value (integer)

©2013-2022 Tencent Cloud. All rights reserved.

Operation	Feature Description
FIELDSET	Updates the values of one or multiple specified fields
FIELDGET	Gets the values of one or multiple specified fields
INDEXGET	Queries index by specified index and field
TRAVERSE	Traverses full table by specified table name

TcaplusDB SDK Conventions

You need to define table information such as primary key fields for TcaplusDB. The extension `tcaplusservice.optionv1.proto` is embedded in the TcaplusDB system, so you only need to import it when customizing a table but do not need to upload it when creating a table or modifying an existing table. Its specific content is as shown below:

```
extend google.protobuf.MessageOptions
{
optional string tcaplus_primary_key = 60000; // Define the table primary key
repeated string tcaplus_index = 60001; // Define the table index
optional string tcaplus_field_cipher_suite = 60002; // Define the encryption algo
rithm used by table fields
optional string tcaplus_record_cipher_suite = 60003; // Define the encryption alg
orithm used by table fields, which is reserved now
optional string tcaplus_cipher_md5 = 60004; // Return the information summary of
the `cipher` field
optional string tcaplus_sharding_key = 60005; // TcaplusDB sharding key
}

extend google.protobuf.FieldOptions
{
optional uint32 tcaplus_size = 60000; // Field size, which is reserved now
optional string tcaplus_desc = 60001; // Field description
optional bool tcaplus_crypto = 60002; // Whether to encrypt the field. The encryp
tion algorithm is defined by `tcaplus_field_cipher_suite`
}
```

The complete file can be found in the `release\x86_64\include\tcaplus_pb_api\tcaplusservice.optionv1.proto` directory.

1. The table name must start with a letter or underscore and can only contain up to 31 digits, letters, and underscores.
2. Protobuf has limited that the name of each field can contain only letters or underscores.

©2013-2022 Tencent Cloud. All rights reserved.

3. A primary key can have up to 4 fields, which must be in `required` type, and their length after packaging cannot exceed 1,022 bytes.
4. The size of a packaged field value cannot exceed 256 KB, and the whole packaged record cannot exceed 256 KB.
5. Besides the primary key fields, there must be at least one general key.
6. A table is in `generic` type by default, which is not displayed publicly.
7. Currently, the primary key fields can only be in scalar value types specified by Protobuf rather than other complex or custom types.

Common API Description

- **int Get(::google::protobuf::Message &msg);**

```
@brief This API is used to get multiple record values and input them into the `vec` structure in `res` through the index based on the entered `index` name, `msg` value, `offset`, and `limit` in `req` and return the total number of records and number of remaining records
@param [INOUT] req Enteredd `req`
@param [INOUT] res Entered `res`
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Success
```

- **int BatchGet(std::vector< ::google::protobuf::Message * > &msgs);**

```
@brief This API is used to get the `msg` field values in batches based on the entered `msgs` field values and input them into `msgs`
@param [INOUT] msgs Entered field list, which will return the specified fields and input them into `msgs`
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Success. 0 will be returned only if at least one field is successfully queried.
```

- **int Add(::google::protobuf::Message *msg);**

```
@brief This API is used to insert the `msg` data record based on the field entered in `msg`. If the field already exists, an error will be reported, and the operation will exit
@param [INOUT] msg Entered field value and data record `msg` to be inserted
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Success
```

©2013-2022 Tencent Cloud. All rights reserved.

- **int Set(const ::google::protobuf::Message &msg);**

```
@brief This API is used to update the specified record value based on the field entered in `msg`. If the record already exists, it will be updated to the specified value; otherwise, the specified record will be inserted
@param [INOUT] msg Entered field value and data record `msg` to be set
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Success
```

- **int Del(const ::google::protobuf::Message &msg);**

```
@brief This API is used to delete `msg` based on the field value entered in `msg`
@param [IN] msg Entered field value, which will return the specified fields and input them into `msg`
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Success. 0 will be returned only if at least one field is successfully queried.
```

- **int FieldInc(::google::protobuf::Message &msg, const std::set &dottedpaths);**

```
@brief This API is used to add a field value in `msg` based on the field value entered in `msg`, `values` incremental value, and field name specified by `dottedpaths`. The field is a numeric variable
@param [INOUT] msg Data record `msg`, which contains the entered field value and will return the result value of the incremental field and update it into `msg`
@param [IN] dottedpaths Dotted and nested string set of field names
@retval <0 Failure. The corresponding error code will be returned, which indicates that no fields are updated
@retval 0 All fields are updated successfully
```

- **int FieldGet(const std::set<std::string> &dottedpaths, ::google::protobuf::Message *msg, std::set<std::string> *failedpaths);**

```
@brief This API is used to get the value of a specified field and input it into `msg` based on the field value entered in `msg` and field name specified by `dottedpaths`
@param [INOUT] msg Data record `msg`, which contains the entered field value and will return the specified field and input it into `msg`
@param [IN] dottedpaths Dotted and nested string set of field names
@param [OUT] failedpaths Dotted and nested string set of field names which fail
```

©2013-2022 Tencent Cloud. All rights reserved.

```
ed to be found will be returned
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Success. 0 will be returned only if at least one field is successfull
y queried
```

- **int FieldSet(::google::protobuf::Message &msg, const std::set &dottedpaths);**

```
@brief This API is used to update the value of a specified field based on the f
ield value entered in `msg` and field name specified by `dottedpaths`. If the v
alue does not exist on the server, it will be inserted
@param [IN] msg Entered field value, which will return the specified fields and
input them into `msg`
@param [IN] dottedpaths Dotted and nested string set of field names
@retval <0 Failure. The corresponding error code will be returned, which indica
tes that no fields are updated
@retval 0 All fields are updated successfully
```

- **int Get(NS_TCAPLUS_PROTOBUF_API::IndexGetRequest& req, NS_TCAPLUS_PROTOBUF_API::IndexGetResponse *res);**

```
@brief This API is used to get multiple record values and input them into the `
vec` structure in `res` through the index based on the entered `index` name, `m
sg` value, `offset`, and `limit` in `req` and return the total number of record
s and number of remaining records
@param [INOUT] req Enteredd `req`
@param [INOUT] res Entered `res`
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Success
```

- **int Traverse(::google::protobuf::Message *msg, TcaplusTraverseCallback *cb);**

```
@brief This API is used to traverse the table. The message will be input into `
msg`
@param [INOUT] msg The specified field will be returned and input into `msg`
@param [INOUT] cb Callback function
@retval <0 Failure. The corresponding error code will be returned
@retval 0 Traversal succeeded
```

Rules and Limits

©2013-2022 Tencent Cloud. All rights reserved.

Get operation limits

- You cannot query a record on a specified version.
- If the original record of the field to be queried does not exist, the field's default value will be returned.

BatchGet operation limits

- One batch query result will be returned for one batch query request, and the batch query timeout period is 10 seconds.
- The number of records in a batch query result is the same as that of the records in the request. Even if a record does not exist or failed to be queried, an empty record will still be output in the result set. Therefore, you need to use `FetchRecord` to get the records.
- The total size of a batch query result set cannot exceed 256 KB; otherwise, only empty records will be returned.
- The sequence of the records returned in the batch query result set may be different from that of the records in the request.

FieldInc operation limits

- Field records specified in `message` must exist.
- Fields specified in `dottedpaths` must be numeric.
- The total number of elements in the `set` where `dottedpaths` is cannot exceed 128.
- The length of each element in the `set` where `dottedpaths` is cannot exceed 1,023 bytes.
- The number of nesting levels of a field represented by each element in the `set` where `dottedpaths` is cannot exceed 32.

FieldGet operation limits

- The total number of elements in the `set` where `dottedpaths` is cannot exceed 128.
- The length of each element in the `set` where `dottedpaths` is cannot exceed 1,023 bytes.
- The number of nesting levels of a field represented by each element in the `set` where `dottedpaths` is cannot exceed 32.

FieldSet operation limits

- The total number of elements in the `set` where `dottedpaths` is cannot exceed 128.
- The length of each element in the `set` where `dottedpaths` is cannot exceed 1,023 bytes.
- The number of nesting levels of a field represented by each element in the `set` where `dottedpaths` is cannot exceed 32.

SDK Source File Directory Structure

©2013-2022 Tencent Cloud. All rights reserved.

```
`-- release
`-- x86_64
|-- docs Documentation directory
| `-- tcaplus
| `-- readme.txt User guide file of SDK for C++
|-- examples Directory of SDK for C++ samples, which include sync and async sampl
es and can be modified and used directly
|-- include Header file directory of SDK for C++
| `-- tcaplus_pb_api Header folder of TcaplusDB PB API
| |-- cipher_suite_base.h Base class of data encryption algorithm suite
| |-- default_aes_cipher_suite.h Default implementation class of data encryption
algorithm suite
| |-- tcaplus_async_pb_api.h Async mode header file, which uses the `TcaplusAsync
PbApi` class in async mode
| |-- tcaplus_coroutine_pb_api.h Coroutine mode header file, which uses the `Tcap
lusCoroutinePbApi` class in coroutine mode
| |-- tcaplus_error_code.h Error code header file, where the corresponding defini
tions and descriptions of all involved error codes can be found
| |-- tcaplus_protobuf_api.h Aggregated API header file, which contains other hea
der files. You can simply import this file for easy development
| |-- tcaplus_protobuf_define.h Basic structure and macro definition header file,
which contains the `ClientOptions` structure and `MESSAGE_OPTION_*` macro definit
ion
| |-- tcaplusservice.optionv1.pb.h Protobuf header file for common TcaplusDB tabl
e definition
| `-- tcaplusservice.optionv1.proto .proto source file for common TcaplusDB table
definition. This file is required when you customize a table
|-- lib Library file directory of SDK for C++
| `-- libtcaplusprotobufapi.a Library file of SDK for C++, which must be containe
d in the program's final link library
`-- version Version record file of SDK for C++
```

©2013-2022 Tencent Cloud. All rights reserved.

TcaplusDB Error Codes

Last updated：2021-01-04 14:22:00

No.	Error Code	Description
1	-1792	The table is read-only. Please check if the usage of RCUs, WCUs, or storage capacity exceeds the threshold set by users.
3	261	The record does not exist.
4	-525	The request for `batchget` timed out.
5	-781	The `batchget` , `getbypartkey` , or `listgetall` method returns more response data than the upper limit (i.e., 1 MB).
6	-1037	The system is busy.
7	-1293	The record already exists. Please do not insert it again.
8	-1549	The accessed table field does not exist.
9	-2061	Incorrect table field type
10	-3085	An incorrect field is specified in the `SetFieldName` operation.
11	-3341	The field value exceeds the maximum size defined by its type.
12	-4109	The subscript of a LIST element is out of range.
14	-4621	The request has no primary key field or index field.
15	-6157	The LIST table has more elements than the defined upper limit. Please enable element removal.
16	-6925	Incorrect `result\_flag` setting. Please refer to the `result\_flag` instructions in the SDK.
17	-7949	Please check the optimistic locking. The requested record version number is inconsistent with the actual record version number.
18	-11277	The method to manipulate the table does not exist.
19	-16141	Failed to perform the `GetRecord` operation on the Protobuf table.
20	-16397	The value of the non-primary key field in the Protobuf table exceeds the maximum size (i.e.,

©2013-2022 Tencent Cloud. All rights reserved.

		256 KB).
21	-16653	Failed to perform the `FieldSetRecord` operation on the Protobuf table.
22	-16909	Failed to perform the `FieldIncRecord` operation on the Protobuf table.
23	-275	The maximum number of primary key fields has been reached. A GENERIC table can have up to 4 primary key fields, and a LIST table 3.
24	-531	The maximum number of non-primary key fields has been reached. A GENERIC table can have up to 128 non-primary key fields, and a LIST table 127.
25	-787	The field name exceeds the maximum size (i.e., 32 B).
26	-1043	The field value exceeds the maximum size (i.e., 256 KB).
27	-1555	The data type of the field value is inconsistent with the defined type.
28	-5395	The request has no primary key.
29	-9235	The index does not exist.
30	-12307	Failed to send the request due to network overload.
31	-12819	The table does not exist.
32	-13843	Failed to send the request due to backend network errors.
33	-14099	The inserted record exceeds the maximum size (1 MB).

©2013-2022 Tencent Cloud. All rights reserved.

SDK Installation
Installation Guide for Linux

Last updated：2021-09-26 17:11:04

Operation Scenarios

This document describes how to install and use a TcaplusDB PB table on Linux.

Runtime Environment

Linux 2.6 and SUSE 12 64-bit.

Prerequisites

Before installing and using TcaplusDB PB, you need to install Protobuf. Currently, TcaplusDB supports Protobuf v2.6.1 and v3.5.0.
Protobuf is a cross-language lightweight structured data storage format launched by Google. TcaplusDB supports defining data tables with Protobuf definition files (.proto). Before using a TcaplusDB PB API, you need to install Protobuf on your development server first. You are recommended to use the source code to install Protobuf in the following steps:

1. Prepare the CVM instance environment.
   You need a server on which CentOS 6 or 7 on the x86_64 architecture has been installed. To compile and construct Protobuf, you also need to install the following programs accordingly:
   - Autoconf
   - Automake
   - Libtool
   - cURL (used to download Google Mock)
   - Make
   - g++
   - Unzip
2. Download the Protobuf source code installation package. For more information, please see SDK Download.
3. Please install an appropriate version of Protobuf based on your actual needs. The following steps use Protobuf 2.6.1 as an example:

©2013-2022 Tencent Cloud. All rights reserved.

4. Run the following command to decompress the source code installation package and enter the source code root directory:

```
tar -xzvf protobuf-2.6.1.tar.gz
cd ./protobuf-2.6.1
```

5. Configure and specify the installation path prefix. You are recommended to install it under `/usr/local/protobuf`.

```
./configure --prefix=/usr/local/protobuf
```

6. Compile and install it.

```
make
make check
make install
```

7. Perform the test. Run the `protoc` command to check whether the installation succeeded. If the following information is displayed, the installation is successful.

```
# protoc --version
libprotoc 2.6.1
```

Directions

Step 1: Installing TcaplusDB SDK

Download the TcaplusDB SDK onto the development server and run the following command to install the file to the specified installation directory:

```
tar -xzf <installation package path> -C <installation directory>,
```

Step 2: Verifying

After the installation, the root directory structure is as shown below:

Directory and File	Description
include/tcaplus_service/	TcaplusDB service API header file
lib/libtcaplusserviceapi.a	TcaplusDB service API library file
include/tcaplus_service/protobuf/	Protobuf API header file

©2013-2022 Tencent Cloud. All rights reserved.

Directory and File	Description
lib/libtcaplusprotobufapi.a	TcaplusDB Protobuf API library file
examples/tcaplus/ProtoBuf	TcaplusDB Protobuf API sample

Step 3: Running `example` and accessing TcaplusDB

You can develop the corresponding data access logic on the GameSvr game server by referring to API samples in `example`.

1. Decompress the TcaplusDB PB API release package.

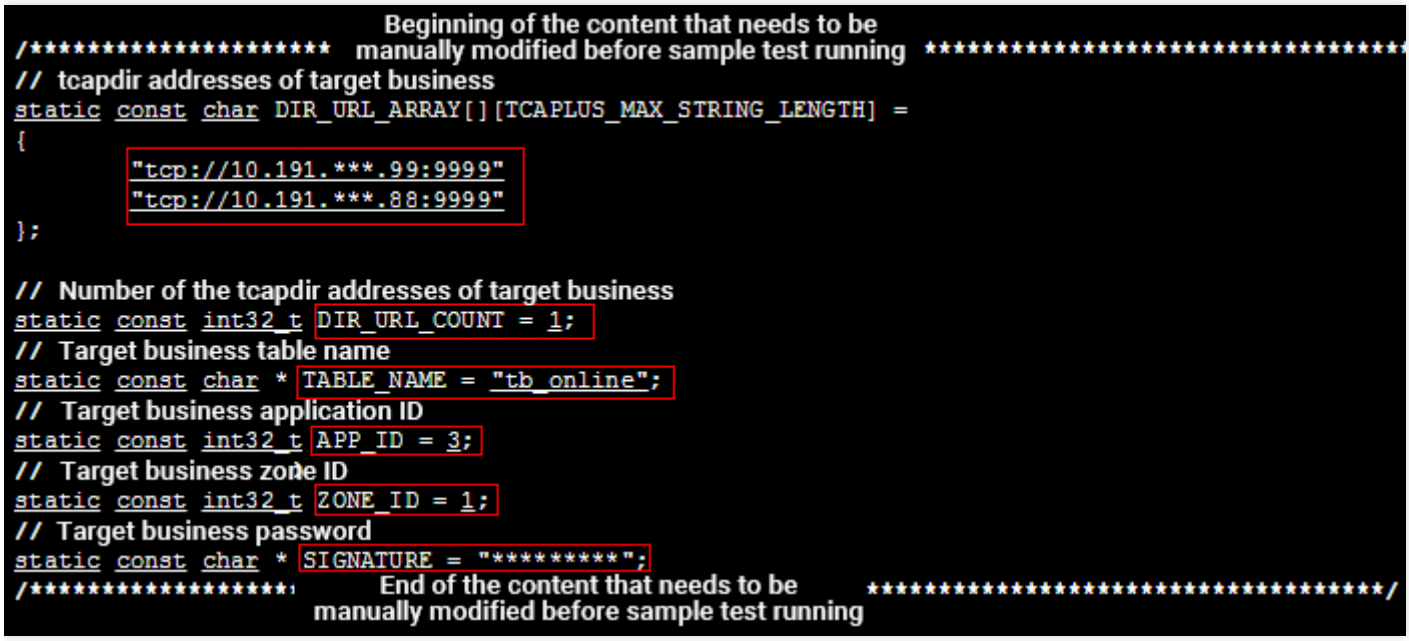
```
tar -xzvf TcaplusPbApi3.36.0.152096.x86_64_release_20170712.tar.gz
```

2. Configure the TcaplusDB system connection information.
   i. Enter the following code on the command line to enter the directory:

```
cd TcaplusPbApi3.36.0.152096.x86_64_release_20170712/release/x86_64/examples/tcaplus/C++_common_for_pb2
```

   ii. On the command line, enter `vi common.h` to modify the `common.h` header file. Modify the content in the figure below based on your actual business needs:
   DIR_URL_ARRAY: cluster access IP address and port.
   DIR_URL_COUNT: value fixed at 1.
   TABLE_NAME: target table to be accessed.
   APP_ID: target access ID.
   ZONE_ID: ID of the table group to be accessed.
   SIGNATURE: cluster access password.

©2013-2022 Tencent Cloud. All rights reserved.



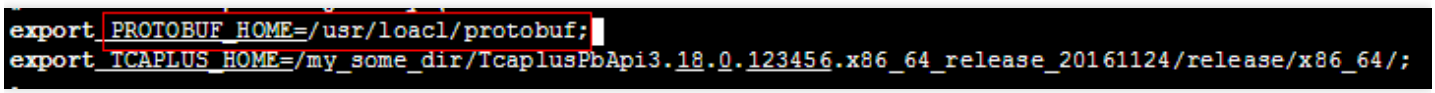
3. Modify the environment configuration file.

In the `TcaplusPbApi3.36.0.152096.x86_64_release_20170712/release/x86_64/examples/tcaplus` directory, there are sample files of calling APIs asynchronously or by using a coroutine. This step uses calling the `Set` API with a coroutine to set data as an example.

Enter the following code on the command line to enter the directory:

```
cd TcaplusPbApi3.18.0.152096.x86_64_release_20170712/release/x86_64/examples/tcaplus/C++_pb2_coroutine_simpletable/SingleOperation/set
```

All code of the example of calling the `Set` API with a coroutine is in the directory. Modify the `envcfg.env` file, set the `PROTOBUF_HOME` environment variable to the Protobuf installation path on this server (specified by `--prefix`), and set the `TCAPLUS_HOME` environment variable to the absolute path of the `release/x86_64` directory in the TcaplusDB PB API package as shown below:



4. Run the following command under the code directory to set the environment variables:

```
source envcfg.env
bash conv.sh
```

5. Compile the binary program.

Run the `make` command to compile the binary program of `example`. After the compilation succeeded, a

©2013-2022 Tencent Cloud. All rights reserved.

`mytest` executable file will be generated.

```
total 32
-rw-r--r-- 1 1002 users  416 Jul 12 16:40 conv.sh
-rw-r--r-- 1 1002 users  232 Aug 15 16:04 envcfg.env
-rw-r--r-- 1 1002 users 1766 Jul 12 16:40 main.cpp
-rw-r--r-- 1 1002 users  678 Jul 12 16:40 Makefile
drwxr-xr-x 2 root root  4096 Aug 15 16:05 mytest
-rw-r--r-- 1 1002 users 1047 Jul 12 16:40 readme.txt
-rw-r--r-- 1 1002 users  707 Jul 12 16:40 table_test.proto
-rw-r--r-- 1 1002 users  662 Jul 12 16:40 tlogconf.xml
```

6. Run the binary program.

   On the command line, enter `./mytest` to run the binary program. The execution result will be displayed in the command line standard output. If any error occurs, please check the `tcaplus_pb.log` log file in the code directory.

©2013-2022 Tencent Cloud. All rights reserved.

Installation Guide for Windows

Last updated：2020-11-20 14:43:40

Operation Scenarios

This document describes how to manipulate TcaplusDB Protobuf API on Windows x64.

Runtime Environment

- OS: Microsoft Windows x86_64
- Compilation environment: Microsoft Visual Studio 2015 (VC14.0)

Directions

Downloading program package

1. Download the dependency package and TcaplusDB Protobuf API program package. For more information, please see SDK Download.
2. Decompress the packages. The program package structure is as shown below:

```
Tcaplus_PbAPI_3.32.0.171987_Win64Vc14MT_Release_20180413
|-- cfg # Configuration directory
|-- docs # Documentation directory
| `-- tcaplus
|-- include # Dependency header file directory
| `-- tcaplus_pb_api
|-- lib # Library directory
| |-- Debug
| `-- Release
|-- examples # Sample directory
`-- tcaplus
|-- C++_common_for_pb2 # Directory of sample common header files
|-- C++_pb2_asyncmode_simpletable # Sample simple PB table in async mode
|-- C++_pb2_coroutine_simpletable # Sample simple PB table in coroutine mode
```

Preparing environment

©2013-2022 Tencent Cloud. All rights reserved.

1. Please make sure that you have enabled the TcaplusDB service in the TcaplusDB Console and obtained the corresponding application information (such as the `AppId`, `ZoneId`, and `AppKey`).
2. Decompress the dependency package and install the dependencies. This document uses the `TSF4G_BASE-2.7.28.164975_Win64Vc14Mt_Release.zip` dependency package as an example. You should download an appropriate dependency package here as needed.
   Supposing the root path for installation is `D:\Tencent\tsf4gMT`, relevant files will be installed under the `D:\Tencent\tsf4gMT\win64vc14MT` path.
3. Compile and install `Porotbuf-3.5.1`.
   - Source code address
   - Compilation and Installation Guide
   - Suppose the installation path is `D:\protobuf-3.5.1`
4. Compile and install `OpenSSL-1.1.0f`.
   - Source code address
   - Compilation and Installation Guide
   - Suppose the installation path is `D:\openssl-1.1.0f`
5. Set environment variables.
   - `TSF4G_HOME="D:\Tencent\tsf4gMT"`
   - `PROTOBUF_HOME="D:\protobuf-3.5.1"`
   - `OPENSSL_HOME="D:\openssl-1.1.0f"`

Constructing

1. Decompress the TcaplusDB PB API installation package.
2. Set the `App` information in the `examples/tcaplus/C++_common_for_pb2/common.h` file.
   - DIR_URL_ARRAY: Tcapdir access point address list
   - DIR_URL_COUNT: number of Tcapdir access point addresses
   - TABLE_NAME: table name. Before using this feature, you need to use the `table_test.xml` file in the `examples` directory to create a `TABLE_NAME` table
   - APP_ID: business ID
   - ZONE_ID: regional business server ID
   - SIGNATURE: business password

```
//examples/tcaplus/C++_common_for_pb2/common.h
/*****************User-defined*****************************/
// Tcapdir access point address list
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.xx.xx.21:9999"
};
// Number of Tcapdir access point addresses
```

©2013-2022 Tencent Cloud. All rights reserved.

```
static const int32_t DIR_URL_COUNT = 1;
// Table name
static const char * TABLE_NAME = "tb_online";
// Business ID
static const int32_t APP_ID = 4;
// Regional business server ID
static const int32_t ZONE_ID = 1;
// Business password
static const char * SIGNATURE = "8e24269ba91fxxxa7e89b1cbb77368e";
/*****************User-defined*****************************/
```

3. Use `examples\tcaplus\C++_pb2_coroutine_simpletable\SingleOperation\set` as an example.

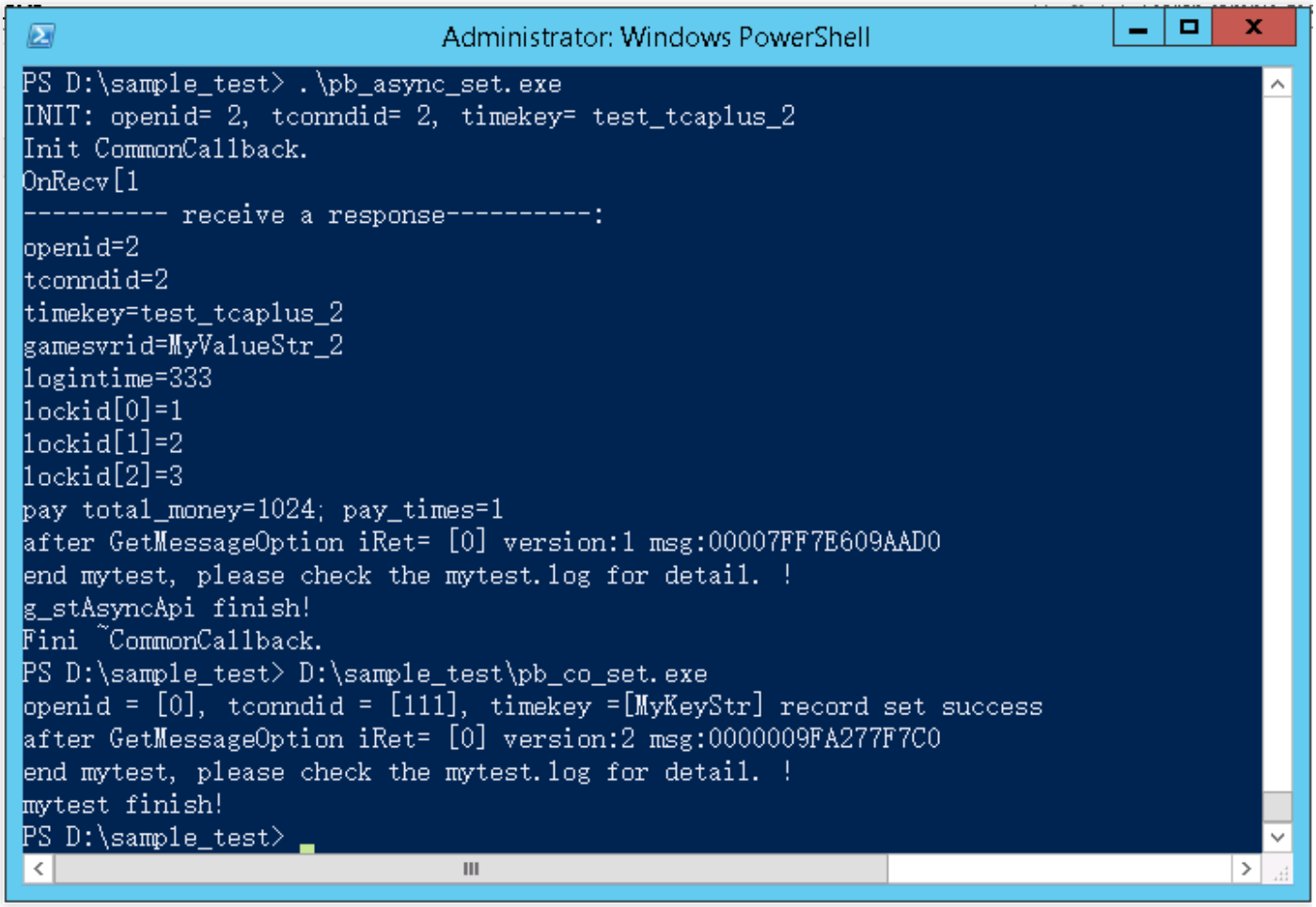
```
set/
|-- main.cpp # Sample code of main function
|-- readme.txt
|-- table_test.proto # .proto table definition file for Tcaplus. The table need
s to be created in advance
|-- pb_co_set.sln # VisualStudio solution file of the project
|-- pb_co_set.vcxproj # VisualStudio project file of the project
|-- proto_generate.cmd # .proto file compilation script
`-- tlogconf.xml
```

   i. First, make sure that you have successfully created a table in the target application by using `table_test.proto`.
   ii. Run the `proto_generate.cmd` script and generate the dependent files under the current path.
      - `table_test.pb.cc`
      - `table_test.pb.h`
   iii. Open the project file `pb_co_set.sln` in Microsoft Visual Studio 2015.
   iv. Generate the solution.
   v. If no errors occur, an executable file `pb_co_set.exe` will be generated under the `examples\tcaplus\C++_pb2_coroutine_simpletable\SingleOperation\set/x64` path.

Testing

1. Copy the `pb_co_set.exe` and `tlogconf.xml` files into the same directory.
2. Switch to the `administrator` role and use `cmd.exe` or `powershell.exe` to run the executable file `pb_co_set.exe`.
3. Check the output.
4. For detailed running information, please check the log file `tcaplus_pb.log`.
5. To run the `*_crypto` sample code, please make sure that the `libcrypto-1_1-x64.dll` file is in the OpenSSL compilation directory under the system path.

©2013-2022 Tencent Cloud. All rights reserved.



TcaplusDB PB API Command List

TcaplusDB PB API provides various types of operations and supports async and coroutine modes. You can find the corresponding usage in the sample code. The list of TcaplusDB PB API commands are as shown below:

Command	Description
SET	Sets record by specifying all its primary keys. If the record already exists, an overwrite operation will be performed; otherwise, an insert operation will be performed.
GET	Queries record by specifying all its primary keys in TcaplusDB PB table. If the record does not exist, an error will be returned.
ADD	Inserts record by specifying all its primary keys. If the record already exists, an error will be returned.
DELETE	Deletes record by specifying all its primary keys. If the record does not exist, an error will be returned.

©2013-2022 Tencent Cloud. All rights reserved.

BATCHGET	Queries multiple records by specifying multiple set of primary keys in TcaplusDB PB table.
TRAVERSE	Traverses TcaplusDB PB table. Multiple records will be returned.
FIELDGET	Queries record by specifying all its primary keys in TcaplusDB PB table. This operation queries and transfers only the values of specified fields, which reduces the network transfer traffic. If the record does not exist, an error will be returned.
FIELDSET	Sets specified fields by specifying all primary keys of record. Only the values of specified fields will be transferred, which reduces the network traffic. If the data record already exists, an update operation will be performed; otherwise, an error will be returned.
FIELDINC	Auto-increments specified fields by specifying all primary keys of record. This command supports only fields in `int32`, `int64`, `uint32`, and `uint64` types. It is similar to `FIELDSET`.
GETBYPARTKEY	Queries multiple eligible data entries by specifying certain primary keys. The specified primary key set must have an index created when the table is created; otherwise, an error will be returned.

©2013-2022 Tencent Cloud. All rights reserved.

Directions for Protobuf Table SDK for C++ Writing Data

Last updated：2021-01-04 14:22:00

Prerequisites

You have created a cluster, a table group, and the `tb_online` table.

The description file table_test.proto of the `tb_online` table is as follows (tcaplusservice.optionv1.proto is the dependent table):

```
syntax = "proto2";

package myTcaplusTable;

import "tcaplusservice.optionv1.proto";

message tb_online {
option(tcaplusservice.tcaplus_primary_key) = "openid,tconndid,timekey";

required int32 openid = 1; //QQ Uin
required int32 tconndid = 2;
required string timekey = 3;
required string gamesvrid = 4;
optional int32 logintime = 5 [default = 1];
repeated int64 lockid = 6 [packed = true]; // The `repeated` fields should use th
e `packed` modifier.
optional pay_info pay = 7;

message pay_info {
optional uint64 total_money = 1;
optional uint64 pay_times = 2;
}
}
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
```

©2013-2022 Tencent Cloud. All rights reserved.

```
{
"tcp://10.191.***.99:9999",
"tcp://10.191.***.88:9999"
};
// The number of target cluster addresses
static const int32_t DIR_URL_COUNT = 2;
// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "tb_online" of the target business
static const char * TABLE_NAME = "tb_online";
```

Step 2: Initialize the TcaplusDB API client for Protobuf

```
// TcaplusDB API client for Protobuf
TcaplusAsyncPbApi g_stAsyncApi;
int32_t InitAsyncPbApi()
{
// TcaplusDB API for Protobuf configuration
ClientOptions cfg;
cfg.app_id = APP_ID;
cfg.zones.push_back(ZONE_ID);
strcpy(cfg.signature, SIGNATURE);
for (int32_t i = 0; i < DIR_URL_COUNT; i++)
{
cfg.dirs.push_back(DIR_URL_ARRAY[i]);
}
// The Protobuf table to be accessed
cfg.tables.push_back(TABLE_NAME);
// Log configuration
strncpy(cfg.log_cfg, "tlogconf.xml", sizeof(cfg.log_cfg));
// Timeout period for connection initialization: 5 seconds
cfg.timeout = 5000;

// Initialize connection
int32_t iRet = g_stAsyncApi.Init(cfg);
if (0 != iRet)
{
cout << "ERROR: g_stAsyncApi.Init failed, log cfg: " << cfg.log_cfg << ", iRet: "
<< iRet << "." << endl;
return iRet;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}
return iRet;
}
```

Step 3. Define asynchronous callbacks

```
// Count the number of received response messages
uint32_t g_dwTotalRevNum = 0;
class CommonCallback : public TcaplusPbCallback
{
public:
CommonCallback()
{
cout << "Init CommonCallback." << endl;
}

~CommonCallback()
{
cout << "Fini ~CommonCallback." << endl;
}
// A callback for the received response message, and "msgs" is an array of the re
ceived records
int OnRecv(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnRecv[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

cout << "---------- receive a response----------:" << endl;
cout << "openid=" << t->openid() << endl;
cout << "tconndid=" << t->tconndid() << endl;
cout << "timekey=" << t->timekey() << endl;
cout << "gamesvrid=" << t->gamesvrid() << endl;
cout << "logintime=" << t->logintime() << endl;
for (int32_t i = 0; i < t->lockid_size();i++)
{
cout << "lockid[" << i << "]=" << t->lockid(i) << endl;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}
tb_online_pay_info pay = t->pay();
cout << "pay total_money=" << pay.total_money() << "; pay_times=" << pay.pay_time
s() << endl;

// Get the version of the record
std::string version;
int iRet = g_stAsyncApi.GetMessageOption(*t, NS_TCAPLUS_PROTOBUF_API::MESSAGE_OPT
ION_DATA_VERSION, &version);
cout << "after GetMessageOption iRet= [" << iRet << "] version:" << version.c_str
() << " msg:" << t << endl;
}

return 0;
}

// A callback for a response error
int OnError(const std::vector< ::google::protobuf::Message *> &msgs, int errorcod
e)
{
cout << "OnError[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

if (TcapErrCode::TXHDB_ERR_RECORD_NOT_EXIST == errorcode)
{
cout << "ERROR: openid= " << t->openid() << ", tconndid= " << t->tconndid() << ",
timekey= " << t->timekey() << ", record not exists" << endl;
}
cout << "ERROR: openid = [" << t->openid() << "], tconndid = [" << t->tconndid()
<< "], timekey =[" << t->timekey() << "] failed:%d" << errorcode << endl;
}

return 0;
}

// A callback for a timed-out message
int OnTimeout(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnTimeout[" << msgs.size() << endl;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "TIMEOUT: msgs[" << idx << "] not tb_online type!" << endl;
return -1;
}

cout << "TIMEOUT: openid = [" << t->openid() << "], tconndid = [" << t->tconndid
() << "], timekey =[" << t->timekey() << "] timeout" << endl;
}

return 0;
}

int OnFinish(const NS_TCAPLUS_PROTOBUF_API::MsgParam &param)
{
cout << "OnFinish: " << param.m_nOperation << " req: " << param.m_vecMsgs.size()<
< endl;
return 0;
}
};
```

Step 4. Send the add request

```
int SendAddRequest()
{
static tb_online t;
t.set_openid(2);
t.set_tconndid(2);
t.set_timekey("test_tcaplus_2");
t.set_gamesvrid("MyValueStr_2");
t.set_logintime(333);
for (int32_t i = 0; i < TOTAL_V3_ARRAY_NUM;i++)
{
t.add_lockid(i+1);
}

tb_online_pay_info* pay = new tb_online_pay_info();
pay->set_total_money(1024);
pay->set_pay_times(1);
t.set_allocated_pay(pay);
```

©2013-2022 Tencent Cloud. All rights reserved.

```
cout << "INIT: openid= " << t.openid() << ", tconndid= " << t.tconndid() << ", ti
mekey= " << t.timekey() << endl;

std::string version = "-1";
g_stAsyncApi.SetMessageOption(t, NS_TCAPLUS_PROTOBUF_API::MESSAGE_OPTION_DATA_VER
SION, version);

static CommonCallback cb;
int32_t iRet = g_stAsyncApi.Add(&t, &cb);
if (iRet != TcapErrCode::GEN_ERR_SUC)
{
cout << "ERROR: openid= " << t.openid() << ", tconndid= " << t.tconndid() << ", t
imekey= " << t.timekey() << ", Set Error iRet = " << iRet << endl;
return -1;
}
return 0;
}
```

Samples

Please refer to main.cpp.

```
int main(void) {

// Initialize the API client
int ret = InitAsyncPbApi();
if ( ret != 0)
{
printf("InitAsyncPbApi failed\n");
return -1;
}

// Send the request
ret = SendAddRequest();
if (0 != ret)
{
printf("SendAddRequest failed\n");
return -1;
}

// Receive the response
do
{
// Update and receive the response packet
```

©2013-2022 Tencent Cloud. All rights reserved.

```
g_stAsyncApi.UpdateNetwork();
usleep(1000 * 10);
} while (g_dwTotalRevNum != 1);
return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Reading Data

Last updated：2021-01-04 14:22:00

Prerequisites

You have created a cluster, a table group, and the `tb_online` table.

The description file table_test.proto of the `tb_online` table is as follows (tcaplusservice.optionv1.proto is the dependent table):

```
syntax = "proto2";

package myTcaplusTable;

import "tcaplusservice.optionv1.proto";

message tb_online {
option(tcaplusservice.tcaplus_primary_key) = "openid,tconndid,timekey";

required int32 openid = 1; //QQ Uin
required int32 tconndid = 2;
required string timekey = 3;
required string gamesvrid = 4;
optional int32 logintime = 5 [default = 1];
repeated int64 lockid = 6 [packed = true]; // The `repeated` fields should use th
e `packed` modifier.
optional pay_info pay = 7;

message pay_info {
optional uint64 total_money = 1;
optional uint64 pay_times = 2;
}
}
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.191.***.99:9999",
```

©2013-2022 Tencent Cloud. All rights reserved.

```
"tcp://10.191.***.88:9999"
};
// The number of target cluster addresses
static const int32_t DIR_URL_COUNT = 2;

// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "tb_online" of the target business
static const char * TABLE_NAME = "tb_online";
```

Step 2: Initialize the TcaplusDB API client for Protobuf

```
// TcaplusDB API client for Protobuf
TcaplusAsyncPbApi g_stAsyncApi;
int32_t InitAsyncPbApi()
{
// TcaplusDB API for Protobuf configuration
ClientOptions cfg;
cfg.app_id = APP_ID;
cfg.zones.push_back(ZONE_ID);
strcpy(cfg.signature, SIGNATURE);
for (int32_t i = 0; i < DIR_URL_COUNT; i++)
{
cfg.dirs.push_back(DIR_URL_ARRAY[i]);
}
// The Protobuf table to be accessed
cfg.tables.push_back(TABLE_NAME);
// Log configuration
strncpy(cfg.log_cfg, "tlogconf.xml", sizeof(cfg.log_cfg));
// Timeout period for connection initialization: 5 seconds
cfg.timeout = 5000;

// Initialize connection
int32_t iRet = g_stAsyncApi.Init(cfg);
if (0 != iRet)
{
cout << "ERROR: g_stAsyncApi.Init failed, log cfg: " << cfg.log_cfg << ", iRet: "
<< iRet << "." << endl;
return iRet;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
return iRet;
}
```

Step 3. Define asynchronous callbacks

```
// Count the number of received response messages
uint32_t g_dwTotalRevNum = 0;
class CommonCallback : public TcaplusPbCallback
{
public:
CommonCallback()
{
cout << "Init CommonCallback." << endl;
}

~CommonCallback()
{
cout << "Fini ~CommonCallback." << endl;
}
// A callback for the received response message, and "msgs" is an array of the re
ceived records
int OnRecv(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnRecv[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

cout << "---------- receive a response----------:" << endl;
cout << "openid=" << t->openid() << endl;
cout << "tconndid=" << t->tconndid() << endl;
cout << "timekey=" << t->timekey() << endl;
cout << "gamesvrid=" << t->gamesvrid() << endl;
cout << "logintime=" << t->logintime() << endl;
for (int32_t i = 0; i < t->lockid_size();i++)
{
cout << "lockid[" << i << "]=" << t->lockid(i) << endl;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
tb_online_pay_info pay = t->pay();
cout << "pay total_money=" << pay.total_money() << "; pay_times=" << pay.pay_time
s() << endl;

// Get the version of the record
std::string version;
int iRet = g_stAsyncApi.GetMessageOption(*t, NS_TCAPLUS_PROTOBUF_API::MESSAGE_OPT
ION_DATA_VERSION, &version);
cout << "after GetMessageOption iRet= [" << iRet << "] version:" << version.c_str
() << " msg:" << t << endl;
}

return 0;
}

// A callback for a response error
int OnError(const std::vector< ::google::protobuf::Message *> &msgs, int errorcod
e)
{
cout << "OnError[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

if (TcapErrCode::TXHDB_ERR_RECORD_NOT_EXIST == errorcode)
{
cout << "ERROR: openid= " << t->openid() << ", tconndid= " << t->tconndid() << ",
timekey= " << t->timekey() << ", record not exists" << endl;
}
cout << "ERROR: openid = [" << t->openid() << "], tconndid = [" << t->tconndid()
<< "], timekey =[" << t->timekey() << "] failed:%d" << errorcode << endl;
}

return 0;
}

// A callback for a timed-out message
int OnTimeout(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnTimeout[" << msgs.size() << endl;
for (size_t idx = 0; idx < msgs.size(); idx++)
```

©2013-2022 Tencent Cloud. All rights reserved.

```
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "TIMEOUT: msgs[" << idx << "] not tb_online type!" << endl;
return -1;
}

cout << "TIMEOUT: openid = [" << t->openid() << "], tconndid = [" << t->tconndid
() << "], timekey =[" << t->timekey() << "] timeout" << endl;
}

return 0;
}

int OnFinish(const NS_TCAPLUS_PROTOBUF_API::MsgParam &param)
{
cout << "OnFinish: " << param.m_nOperation << " req: " << param.m_vecMsgs.size()<
< endl;
return 0;
}
};
```

Step 4. Send the get request

```
int SendGetRequest()
{
static tb_online t;
t.set_openid(1);
t.set_tconndid(1);
t.set_timekey("test_tcaplus");

cout << "INIT: openid= " << t.openid() << ", tconndid= " << t.tconndid() << ", ti
mekey=" << t.timekey() << endl;

static CommonCallback cb;
int32_t iRet = g_stAsyncApi.Get(&t, &cb);
if (iRet != TcapErrCode::GEN_ERR_SUC)
{
cout << "ERROR: openid= " << t.openid() << ", tconndid= " << t.tconndid() << ", t
imekey= " << t.timekey() << ", Get Error iRet = " << iRet << endl;
return -1;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
return 0;
}
```

Samples

```
int main(void) {

// Initialize the API client
int ret = InitAsyncPbApi();
if ( ret != 0)
{
printf("InitAsyncPbApi failed\n");
return -1;
}

// Send the request
ret = SendGetRequest();
if (0 != ret)
{
printf("SendSetRequest failed\n");
return -1;
}

// Receive the response
do
{
// Update and receive the response packet
g_stAsyncApi.UpdateNetwork();
usleep(1000 * 10);
} while (g_dwTotalRevNum != 1);
return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Updating Data

Last updated：2021-01-04 14:22:00

Prerequisites

You have created a cluster, a table group, and the `tb_online` table.

The description file table_test.proto of the `tb_online` table is as follows (tcaplusservice.optionv1.proto is the dependent table):

```
syntax = "proto2";

package myTcaplusTable;

import "tcaplusservice.optionv1.proto";

message tb_online {
option(tcaplusservice.tcaplus_primary_key) = "openid,tconndid,timekey";

required int32 openid = 1; //QQ Uin
required int32 tconndid = 2;
required string timekey = 3;
required string gamesvrid = 4;
optional int32 logintime = 5 [default = 1];
repeated int64 lockid = 6 [packed = true]; // The `repeated` fields should use th
e `packed` modifier.
optional pay_info pay = 7;

message pay_info {
optional uint64 total_money = 1;
optional uint64 pay_times = 2;
}
}
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.191.***.99:9999",
```

©2013-2022 Tencent Cloud. All rights reserved.

```
"tcp://10.191.***.88:9999"
};
// The number of target cluster addresses
static const int32_t DIR_URL_COUNT = 2;
// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "tb_online" of the target business
static const char * TABLE_NAME = "tb_online";
```

Step 2: Initialize the TcaplusDB API client for Protobuf

```
// TcaplusDB API client for Protobuf
TcaplusAsyncPbApi g_stAsyncApi;
int32_t InitAsyncPbApi()
{
// TcaplusDB API for Protobuf configuration
ClientOptions cfg;
cfg.app_id = APP_ID;
cfg.zones.push_back(ZONE_ID);
strcpy(cfg.signature, SIGNATURE);
for (int32_t i = 0; i < DIR_URL_COUNT; i++)
{
cfg.dirs.push_back(DIR_URL_ARRAY[i]);
}
// The Protobuf table to be accessed
cfg.tables.push_back(TABLE_NAME);
// Log configuration
strncpy(cfg.log_cfg, "tlogconf.xml", sizeof(cfg.log_cfg));
// Timeout period for connection initialization: 5 seconds
cfg.timeout = 5000;

// Initialize connection
int32_t iRet = g_stAsyncApi.Init(cfg);
if (0 != iRet)
{
cout << "ERROR: g_stAsyncApi.Init failed, log cfg: " << cfg.log_cfg << ", iRet: "
<< iRet << "." << endl;
return iRet;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
return iRet;
}
```

Step 3. Define asynchronous callbacks

```
// Count the number of received response messages
uint32_t g_dwTotalRevNum = 0;
class CommonCallback : public TcaplusPbCallback
{
public:
CommonCallback()
{
cout << "Init CommonCallback." << endl;
}

~CommonCallback()
{
cout << "Fini ~CommonCallback." << endl;
}
// A callback for the received response message, and "msgs" is an array of the re
ceived records
int OnRecv(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnRecv[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

cout << "---------- receive a response----------:" << endl;
cout << "openid=" << t->openid() << endl;
cout << "tconndid=" << t->tconndid() << endl;
cout << "timekey=" << t->timekey() << endl;
cout << "gamesvrid=" << t->gamesvrid() << endl;
cout << "logintime=" << t->logintime() << endl;
for (int32_t i = 0; i < t->lockid_size();i++)
{
cout << "lockid[" << i << "]=" << t->lockid(i) << endl;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
tb_online_pay_info pay = t->pay();
cout << "pay total_money=" << pay.total_money() << "; pay_times=" << pay.pay_time
s() << endl;

// Get the version of the record
std::string version;
int iRet = g_stAsyncApi.GetMessageOption(*t, NS_TCAPLUS_PROTOBUF_API::MESSAGE_OPT
ION_DATA_VERSION, &version);
cout << "after GetMessageOption iRet= [" << iRet << "] version:" << version.c_str
() << " msg:" << t << endl;
}

return 0;
}

// A callback for a response error
int OnError(const std::vector< ::google::protobuf::Message *> &msgs, int errorcod
e)
{
cout << "OnError[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

if (TcapErrCode::TXHDB_ERR_RECORD_NOT_EXIST == errorcode)
{
cout << "ERROR: openid= " << t->openid() << ", tconndid= " << t->tconndid() << ",
timekey= " << t->timekey() << ", record not exists" << endl;
}
cout << "ERROR: openid = [" << t->openid() << "], tconndid = [" << t->tconndid()
<< "], timekey =[" << t->timekey() << "] failed:%d" << errorcode << endl;
}

return 0;
}

// A callback for a timed-out message
int OnTimeout(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnTimeout[" << msgs.size() << endl;
for (size_t idx = 0; idx < msgs.size(); idx++)
```

©2013-2022 Tencent Cloud. All rights reserved.

```
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "TIMEOUT: msgs[" << idx << "] not tb_online type!" << endl;
return -1;
}

cout << "TIMEOUT: openid = [" << t->openid() << "], tconndid = [" << t->tconndid
() << "], timekey =[" << t->timekey() << "] timeout" << endl;
}

return 0;
}

int OnFinish(const NS_TCAPLUS_PROTOBUF_API::MsgParam &param)
{
cout << "OnFinish: " << param.m_nOperation << " req: " << param.m_vecMsgs.size()<
< endl;
return 0;
}
};
```

Step 4. Send the set request

```
int SendSetRequest()
{
// Define the Protobuf record structure and assign it a value. The structure can
be converted with proto.
static tb_online t;
t.set_openid(2);
t.set_tconndid(2);
t.set_timekey("test_tcaplus_2");
t.set_gamesvrid("MyValueStr_2");
t.set_logintime(333);
for (int32_t i = 0; i < 3;i++)
{
t.add_lockid(i+1);
}

tb_online_pay_info* pay = new tb_online_pay_info();
pay->set_total_money(1024);
pay->set_pay_times(1);
t.set_allocated_pay(pay);
```

©2013-2022 Tencent Cloud. All rights reserved.

```
cout << "INIT: openid= " << t.openid() << ", tconndid= " << t.tconndid() << ", ti
mekey= " << t.timekey() << endl;

// Specify the version number which is used by optimistic locking
std::string version = "-1"; // std::string version="1"; "-1" indicates no compari
son, and "1" indicates the version number of the server data is 1.
g_stAsyncApi.SetMessageOption(t, NS_TCAPLUS_PROTOBUF_API::MESSAGE_OPTION_DATA_VER
SION, version);

//Pass in the defined callback function which is automatically invoked when a res
ponse message is received.
static CommonCallback cb;
int32_t iRet = g_stAsyncApi.Set(&t, &cb);
if (iRet != TcapErrCode::GEN_ERR_SUC)
{
cout << "ERROR: openid= " << t.openid() << ", tconndid= " << t.tconndid() << ", t
imekey= " << t.timekey() << ", Set Error iRet = " << iRet << endl;
return -1;
}
return 0;
}
```

Samples

```
int main(void) {

// Initialize the API client
int ret = InitAsyncPbApi();
if ( ret != 0)
{
printf("InitAsyncPbApi failed\n");
return -1;
}

// Send the request
ret = SendSetRequest();
if (0 != ret)
{
printf("SendSetRequest failed\n");
return -1;
}

// Receive the response
```

©2013-2022 Tencent Cloud. All rights reserved.

```
do
{
// Update and receive the response packet
g_stAsyncApi.UpdateNetwork();
usleep(1000 * 10);
} while (g_dwTotalRevNum != 1);
return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Deleting Data

Last updated：2021-01-04 14:22:00

Prerequisites

You have created a cluster, a table group, and the `tb_online` table.

The description file table_test.proto of the `tb_online` table is as follows (tcaplusservice.optionv1.proto is the dependent table):

```
syntax = "proto2";

package myTcaplusTable;

import "tcaplusservice.optionv1.proto";

message tb_online {
option(tcaplusservice.tcaplus_primary_key) = "openid,tconndid,timekey";

required int32 openid = 1; //QQ Uin
required int32 tconndid = 2;
required string timekey = 3;
required string gamesvrid = 4;
optional int32 logintime = 5 [default = 1];
repeated int64 lockid = 6 [packed = true]; // The `repeated` fields should use th
e `packed` modifier.
optional pay_info pay = 7;

message pay_info {
optional uint64 total_money = 1;
optional uint64 pay_times = 2;
}
}
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.191.***.99:9999",
```

©2013-2022 Tencent Cloud. All rights reserved.

```
"tcp://10.191.***.88:9999"
};
// The number of target cluster addresses
static const int32_t DIR_URL_COUNT = 2;
// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "tb_online" of the target business
static const char * TABLE_NAME = "tb_online";
```

Step 2: Initialize the TcaplusDB API client for Protobuf

```
// TcaplusDB API client for Protobuf
TcaplusAsyncPbApi g_stAsyncApi;
int32_t InitAsyncPbApi()
{
// TcaplusDB API for Protobuf configuration
ClientOptions cfg;
cfg.app_id = APP_ID;
cfg.zones.push_back(ZONE_ID);
strcpy(cfg.signature, SIGNATURE);
for (int32_t i = 0; i < DIR_URL_COUNT; i++)
{
cfg.dirs.push_back(DIR_URL_ARRAY[i]);
}
// The Protobuf table to be accessed
cfg.tables.push_back(TABLE_NAME);
// Log configuration
strncpy(cfg.log_cfg, "tlogconf.xml", sizeof(cfg.log_cfg));
// Timeout period for connection initialization: 5 seconds
cfg.timeout = 5000;

// Initialize connection
int32_t iRet = g_stAsyncApi.Init(cfg);
if (0 != iRet)
{
cout << "ERROR: g_stAsyncApi.Init failed, log cfg: " << cfg.log_cfg << ", iRet: "
<< iRet << "." << endl;
return iRet;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
return iRet;
}
```

Step 3. Define asynchronous callbacks

```
// Count the number of received response messages
uint32_t g_dwTotalRevNum = 0;
class CommonCallback : public TcaplusPbCallback
{
public:
CommonCallback()
{
cout << "Init CommonCallback." << endl;
}

~CommonCallback()
{
cout << "Fini ~CommonCallback." << endl;
}
// A callback for the received response message, and "msgs" is an array of the re
ceived records
int OnRecv(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnRecv[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

cout << "---------- receive a response----------:" << endl;
cout << "openid=" << t->openid() << endl;
cout << "tconndid=" << t->tconndid() << endl;
cout << "timekey=" << t->timekey() << endl;
cout << "gamesvrid=" << t->gamesvrid() << endl;
cout << "logintime=" << t->logintime() << endl;
for (int32_t i = 0; i < t->lockid_size();i++)
{
cout << "lockid[" << i << "]=" << t->lockid(i) << endl;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
tb_online_pay_info pay = t->pay();
cout << "pay total_money=" << pay.total_money() << "; pay_times=" << pay.pay_time
s() << endl;

// Get the version of the record
std::string version;
int iRet = g_stAsyncApi.GetMessageOption(*t, NS_TCAPLUS_PROTOBUF_API::MESSAGE_OPT
ION_DATA_VERSION, &version);
cout << "after GetMessageOption iRet= [" << iRet << "] version:" << version.c_str
() << " msg:" << t << endl;
}

return 0;
}

// A callback for a response error
int OnError(const std::vector< ::google::protobuf::Message *> &msgs, int errorcod
e)
{
cout << "OnError[" << msgs.size() << endl;
g_dwTotalRevNum++;
for (size_t idx = 0; idx < msgs.size(); idx++)
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "ERROR: msgs[" << idx << "] not tb_online type." << endl;
return -1;
}

if (TcapErrCode::TXHDB_ERR_RECORD_NOT_EXIST == errorcode)
{
cout << "ERROR: openid= " << t->openid() << ", tconndid= " << t->tconndid() << ",
timekey= " << t->timekey() << ", record not exists" << endl;
}
cout << "ERROR: openid = [" << t->openid() << "], tconndid = [" << t->tconndid()
<< "], timekey =[" << t->timekey() << "] failed:%d" << errorcode << endl;
}

return 0;
}

// A callback for a timed-out message
int OnTimeout(const std::vector< ::google::protobuf::Message *> &msgs)
{
cout << "OnTimeout[" << msgs.size() << endl;
for (size_t idx = 0; idx < msgs.size(); idx++)
```

©2013-2022 Tencent Cloud. All rights reserved.

```
{
tb_online* t = dynamic_cast<tb_online *>(msgs[idx]);
if (NULL == t)
{
cout << "TIMEOUT: msgs[" << idx << "] not tb_online type!" << endl;
return -1;
}

cout << "TIMEOUT: openid = [" << t->openid() << "], tconndid = [" << t->tconndid
() << "], timekey =[" << t->timekey() << "] timeout" << endl;
}

return 0;
}

int OnFinish(const NS_TCAPLUS_PROTOBUF_API::MsgParam &param)
{
cout << "OnFinish: " << param.m_nOperation << " req: " << param.m_vecMsgs.size()<
< endl;
return 0;
}
};
```

Step 4. Send the delete request

```
int SendDelRequest()
{
static tb_online t;
t.set_openid(1);
t.set_tconndid(1);
t.set_timekey("test_tcaplus");

static CommonCallback cb;
std::string version = "-1";
g_stApi.SetMessageOption(t, NS_TCAPLUS_PROTOBUF_API::MESSAGE_OPTION_DATA_VERSION,
version);
int32_t iRet = g_stAsyncApi.Del(&t, &cb);
if (iRet != TcapErrCode::GEN_ERR_SUC)
{
cout << "ERROR: openid= " << t.openid() << ", tconndid= " << t.tconndid() << ", t
imekey= " << t.timekey() << ", Delete Error iRet = " << iRet << endl;
return -1;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
return 0;
}
```

Samples

```
int main(void) {

// Initialize the API client
int ret = InitAsyncPbApi();
if ( ret != 0)
{
printf("InitAsyncPbApi failed\n");
return -1;
}

// Send the request
ret = SendDelRequest();
if (0 != ret)
{
printf("SendDelRequest failed\n");
return -1;
}

// Receive the response
do
{
// Update and receive the response packet
g_stAsyncApi.UpdateNetwork();
usleep(1000 * 10);
} while (g_dwTotalRevNum != 1);
return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Directions for TDR Table SDK for C++ Writing Data

Last updated：2021-01-04 14:35:23

Prerequisites

You have created a cluster, a table group and the `PLAYERONLINECNT` table.

The description file table_test.xml of the `PLAYERONLINECNT` table is as follows:

```
<?xml version="1.0" encoding="GBK" standalone="yes" ?>
<metalib name="tcaplus_tb" tagsetversion="1" version="1">
<struct name="PLAYERONLINECNT" version="1" primarykey="TimeStamp,GameSvrID" split
tablekey="TimeStamp">
<entry name="TimeStamp" type="uint32" desc="Measured in minutes" />
<entry name="GameSvrID" type="string" size="64" />
<entry name="GameAppID" type="string" size="64" desc="gameapp id" />
<entry name="OnlineCntIOS" type="uint32" defaultvalue="0" desc="The number of iOS
online users" />
<entry name="OnlineCntAndroid" type="uint32" defaultvalue="0" desc="The number of
Android online users" />
<entry name="BinaryLen" type="smalluint" defaultvalue="1" desc="Data length of th
e data source; skip the source check when the length is 0."/>
<entry name="binary" type="tinyint" desc="Binary" count= "1000" refer="BinaryLen"
/>
<entry name="binary2" type="tinyint" desc="Binary 2" count= "1000" refer="BinaryL
en" />
<entry name="strstr" type="string" size="64" desc="String"/>
<index name="index_id" column="TimeStamp"/>
</struct>
</metalib>
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.191.***.99:9999",
"tcp://10.191.***.88:9999"
```

©2013-2022 Tencent Cloud. All rights reserved.

```
};
// The number of target cluster addresses
static const int32_t DIR_URL_COUNT = 2;
// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "PLAYERONLINECNT" of the target business
static const char * TABLE_NAME = "PLAYERONLINECNT";
```

Step 2: Initialize TcaplusAPI log handles

Log configuration file: tlogconf.xml

```
// TCaplus service log class
TcaplusService::TLogger* g_pstTlogger;
LPTLOGCATEGORYINST g_pstLogHandler;
LPTLOGCTX g_pstLogCtx;
int32_t InitLog()
{
// Absolute path of the log configuration file
const char* sLogConfFile = "tlogconf.xml";
// Log class name
const char* sCategoryName = "mytest";
// Initialize the log handle using the configuration file
g_pstLogCtx = tlog_init_from_file(sLogConfFile);
if (NULL == g_pstLogCtx)
{
fprintf(stderr, "tlog_init_from_file failed.\n");
return -1;
}
// Get the log class
g_pstLogHandler = tlog_get_category(g_pstLogCtx, sCategoryName);
if (NULL == g_pstLogHandler)
{
fprintf(stderr, "tlog_get_category(mytest) failed.\n");
return -2;
}
// Initialize the log handle
g_pstTlogger = new TcaplusService::TLogger(g_pstLogHandler);
if (NULL == g_pstTlogger)
{
```

©2013-2022 Tencent Cloud. All rights reserved.

```
fprintf(stderr, "TcaplusService::TLogger failed.\n");
return -3;
}

return 0;
}
```

Step 3: Initialize TcaplusAPI client

```
// The client main class of the TCaplus service API
TcaplusService::TcaplusServer g_stTcapSvr;
// Meta information for the table
extern unsigned char g_szMetalib_tcaplus_tb[];
LPTDRMETA g_szTableMeta = NULL;
int32_t InitServiceAPI()
{
// Initialize
int32_t iRet = g_stTcapSvr.Init(g_pstTlogger, /*module_id*/0, /*app id*/APP_ID, /
*zone id*/ZONE_ID, /*signature*/SIGNATURE);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.Init failed, iRet: %d.", iRet);
return iRet;
}

// Add the directory server
for (int32_t i = 0; i< DIR_URL_COUNT; i++)
{
iRet = g_stTcapSvr.AddDirServerAddress(DIR_URL_ARRAY[i]);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.AddDirServerAddress(%s) failed, iR
et: %d.", DIR_URL_ARRAY[i], iRet);
return iRet;
}
}

// Get the meta description of the table
g_szTableMeta = tdr_get_meta_by_name((LPTDRMETALIB)g_szMetalib_tcaplus_tb, TABLE_
NAME);
if(NULL == g_szTableMeta)
{
tlog_error(g_pstLogHandler, 0, 0,"tdr_get_meta_by_name(%s) failed.", TABLE_NAME);
return -1;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}

// Register the data table (connect to the directory server, verify the password,
and get the table routing) with 10-second timeout period
iRet = g_stTcapSvr.RegistTable(TABLE_NAME, g_szTableMeta, /*timeout_ms*/10000);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.RegistTable(%s) failed, iRet: %d."
, TABLE_NAME, iRet);
return iRet;
}

// Connect to all TcaplusDB proxy servers corresponding to the table
iRet = g_stTcapSvr.ConnectAll(/*timeout_ms*/10000, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.ConnectAll failed, iRet: %d.", iRe
t);
return iRet;
}

return 0;
}
```

Step 4. Send a replace request via the API

You can also send `TCAPLUS_API_INSERT_REQ` to insert data. If the data already exists, a failure message will be returned.

If you send `TCAPLUS_API_REPLACE_REQ` to insert data, the existing data will be replaced.

```
int32_t SendReplaceRequest()
{
// Request object class
TcaplusService::TcaplusServiceRequest* pstRequest = g_stTcapSvr.GetRequest(TABLE_
NAME);
if (NULL == pstRequest)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.GetRequest(%s) failed.", TABLE_NAM
E);
return -1;
}

// Initialize the request object
```

©2013-2022 Tencent Cloud. All rights reserved.

```
int iRet = pstRequest->Init(TCAPLUS_API_REPLACE_REQ, NULL, 0, 0, 0, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->Init(TCAPLUS_API_REPLACE_REQ) fail
ed, iRet: %d.", iRet);
return iRet;
}

// Add a record to the table according to the request
TcaplusService::TcaplusServiceRecord* pstRecord = pstRequest->AddRecord();
if (NULL == pstRecord)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->AddRecord() failed.");
return -1;
}

PLAYERONLINECNT stPLAYERONLINECNT;
memset(&stPLAYERONLINECNT, 0, sizeof(stPLAYERONLINECNT));

// Specify the information of the key to be updated
stPLAYERONLINECNT.dwTimeStamp = 1;
snprintf(stPLAYERONLINECNT.szGameSvrID, sizeof(stPLAYERONLINECNT.szGameSvrID), "%
s", "mysvrid");

// Specify the information of the value to be updated
snprintf(stPLAYERONLINECNT.szGameAppID, sizeof(stPLAYERONLINECNT.szGameAppID), "%
s", "myappid");
stPLAYERONLINECNT.dwOnlineCntIOS = 1;
stPLAYERONLINECNT.dwOnlineCntAndroid = 1;

// Set the record based on the TDR description
iRet = pstRecord->SetData(&stPLAYERONLINECNT, sizeof(stPLAYERONLINECNT));
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRecord->SetData() failed, iRet: %d.", iRet)
;
return iRet;
}

// Send the request packet
iRet= g_stTcapSvr.SendRequest(pstRequest);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.SendRequest failed, iRet: %d.", iR
et);
return iRet;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
return 0;
}
```

Step 5. Receive the response via the API

```
int RecvResp(TcaplusServiceResponse*& response)
{
// Block the reception of response packets 5,000 times with each block lasting fo
r 1 ms
unsigned int sleep_us = 1000;
unsigned int sleep_count = 5000;
do
{
usleep(sleep_us);
response = NULL;
int ret = g_stTcapSvr.RecvResponse(response);
if (ret < 0)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse failed. ret:%d", r
et);
return ret;
}

// Receive a response packet
if (1 == ret)
{
break;
}
} while ((--sleep_count) > 0);

// Timeout period: 5 seconds
if (0 == sleep_count)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse wait timeout.");
return -1;
}
return 0;
}
```

Samples

©2013-2022 Tencent Cloud. All rights reserved.

Please refer to main.cpp.

```
int main(void) {
// Initialize the log
int ret = InitLog();
if ( ret != 0)
{
printf("init log failed\n");
return -1;
}

// Initialize the API client
ret = InitServiceAPI();
if ( ret != 0)
{
printf("init InitServiceAPI failed\n");
return -1;
}

// Send the request
ret = SendReplaceRequest();
if (0 != ret)
{
printf("SendReplaceRequest failed\n");
return -1;
}

// Receive the response
TcaplusServiceResponse* response = NULL;
ret = RecvResp(response);
if (0 != ret)
{
printf("RecvResp failed\n");
return -1;
}

// Get the result, where "0" indicates success
int32_t result = response->GetResult();
if (0 != result)
{
printf("the result is %d\n", result);
return -1;
}
return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Reading Data

Last updated：2021-01-04 14:35:23

Prerequisites

You have created a cluster, a table group and the `PLAYERONLINECNT` table.

The description file table_test.xml of the `PLAYERONLINECNT` table is as follows:

```
<?xml version="1.0" encoding="GBK" standalone="yes" ?>
<metalib name="tcaplus_tb" tagsetversion="1" version="1">
<struct name="PLAYERONLINECNT" version="1" primarykey="TimeStamp,GameSvrID" split
tablekey="TimeStamp">
<entry name="TimeStamp" type="uint32" desc="Measured in minutes" />
<entry name="GameSvrID" type="string" size="64" />
<entry name="GameAppID" type="string" size="64" desc="gameapp id" />
<entry name="OnlineCntIOS" type="uint32" defaultvalue="0" desc="The number of iOS
online users" />
<entry name="OnlineCntAndroid" type="uint32" defaultvalue="0" desc="The number of
Android online users" />
<entry name="BinaryLen" type="smalluint" defaultvalue="1" desc="Data length of th
e data source; skip the source check when the length is 0."/>
<entry name="binary" type="tinyint" desc="Binary" count= "1000" refer="BinaryLen"
/>
<entry name="binary2" type="tinyint" desc="Binary 2" count= "1000" refer="BinaryL
en" />
<entry name="strstr" type="string" size="64" desc="String"/>
<index name="index_id" column="TimeStamp"/>
</struct>
</metalib>
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.191.***.99:9999",
"tcp://10.191.***.88:9999"
};
// The number of target cluster addresses
```

©2013-2022 Tencent Cloud. All rights reserved.

```
static const int32_t DIR_URL_COUNT = 2;
// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "PLAYERONLINECNT" of the target business
static const char * TABLE_NAME = "PLAYERONLINECNT";
```

Step 2: Initialize TcaplusAPI log handles

Log configuration file: tlogconf.xml

```
// TCaplus service log class
TcaplusService::TLogger* g_pstTlogger;
LPTLOGCATEGORYINST g_pstLogHandler;
LPTLOGCTX g_pstLogCtx;
int32_t InitLog()
{
// Absolute path of the log configuration file
const char* sLogConfFile = "tlogconf.xml";
// Log class name
const char* sCategoryName = "mytest";
// Initialize the log handle using the configuration file
g_pstLogCtx = tlog_init_from_file(sLogConfFile);
if (NULL == g_pstLogCtx)
{
fprintf(stderr, "tlog_init_from_file failed.\n");
return -1;
}
// Get the log class
g_pstLogHandler = tlog_get_category(g_pstLogCtx, sCategoryName);
if (NULL == g_pstLogHandler)
{
fprintf(stderr, "tlog_get_category(mytest) failed.\n");
return -2;
}
// Initialize the log handle
g_pstTlogger = new TcaplusService::TLogger(g_pstLogHandler);
if (NULL == g_pstTlogger)
{
fprintf(stderr, "TcaplusService::TLogger failed.\n");
return -3;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}

return 0;
}
```

Step 3: Initialize TcaplusAPI client

```
// The client main class of the TCaplus service API
TcaplusService::TcaplusServer g_stTcapSvr;
// Meta information for the table
extern unsigned char g_szMetalib_tcaplus_tb[];
LPTDRMETA g_szTableMeta = NULL;
int32_t InitServiceAPI()
{
// Initialize
int32_t iRet = g_stTcapSvr.Init(g_pstTlogger, /*module_id*/0, /*app id*/APP_ID, /
*zone id*/ZONE_ID, /*signature*/SIGNATURE);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.Init failed, iRet: %d.", iRet);
return iRet;
}

// Add the directory server
for (int32_t i = 0; i< DIR_URL_COUNT; i++)
{
iRet = g_stTcapSvr.AddDirServerAddress(DIR_URL_ARRAY[i]);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.AddDirServerAddress(%s) failed, iR
et: %d.", DIR_URL_ARRAY[i], iRet);
return iRet;
}
}

// Get the meta description of the table
g_szTableMeta = tdr_get_meta_by_name((LPTDRMETALIB)g_szMetalib_tcaplus_tb, TABLE_
NAME);
if(NULL == g_szTableMeta)
{
tlog_error(g_pstLogHandler, 0, 0,"tdr_get_meta_by_name(%s) failed.", TABLE_NAME);
return -1;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Register the data table (connect to the directory server, verify the password,
and get the table routing) with 10-second timeout period
iRet = g_stTcapSvr.RegistTable(TABLE_NAME, g_szTableMeta, /*timeout_ms*/10000);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.RegistTable(%s) failed, iRet: %d."
, TABLE_NAME, iRet);
return iRet;
}

// Connect to all TcaplusDB proxy servers corresponding to the table
iRet = g_stTcapSvr.ConnectAll(/*timeout_ms*/10000, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.ConnectAll failed, iRet: %d.", iRe
t);
return iRet;
}

return 0;
}
```

Step 4. Send a get request via the API

```
int32_t SendGetRequest()
{
// Request object class
TcaplusService::TcaplusServiceRequest* pstRequest = g_stTcapSvr.GetRequest(TABLE_
NAME);
if (NULL == pstRequest)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.GetRequest(%s) failed.", TABLE_NAM
E);
return -1;
}

// Initialize the request object
int iRet = pstRequest->Init(TCAPLUS_API_GET_REQ, NULL, 0, 0, 0, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->Init(TCAPLUS_API_GET_REQ) failed,
iRet: %d.", iRet);
return iRet;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Add a record to the table according to the request
TcaplusService::TcaplusServiceRecord* pstRecord = pstRequest->AddRecord();
if (NULL == pstRecord)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->AddRecord() failed.");
return -1;
}

PLAYERONLINECNT stPLAYERONLINECNT;
memset(&stPLAYERONLINECNT, 0, sizeof(stPLAYERONLINECNT));

// Specify the information of the key to be queried
stPLAYERONLINECNT.dwTimeStamp = 1;
snprintf(stPLAYERONLINECNT.szGameSvrID, sizeof(stPLAYERONLINECNT.szGameSvrID), "%
s", "mysvrid");

// Set the record based on the TDR description
iRet = pstRecord->SetData(&stPLAYERONLINECNT, sizeof(stPLAYERONLINECNT));
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRecord->SetData() failed, iRet: %d.", iRet)
;
return iRet;
}

// Send the request packet
iRet= g_stTcapSvr.SendRequest(pstRequest);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.SendRequest failed, iRet: %d.", iR
et);
return iRet;
}
return 0;
}
```

Step 5. Receive the response via the API

```
int RecvResp(TcaplusServiceResponse*& response)
{
// Block the reception of response packets 5,000 times with each block lasting fo
r 1 ms
unsigned int sleep_us = 1000;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
unsigned int sleep_count = 5000;
do
{
usleep(sleep_us);
response = NULL;
int ret = g_stTcapSvr.RecvResponse(response);
if (ret < 0)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse failed. ret:%d", r
et);
return ret;
}

// Receive a response packet
if (1 == ret)
{
break;
}
} while ((--sleep_count) > 0);

// Timeout period: 5 seconds
if (0 == sleep_count)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse wait timeout.");
return -1;
}
return 0;
}
```

Samples

Please refer to main.cpp.

```
int main(void) {
// Initialize the log
int ret = InitLog();
if (ret != 0)
{
printf("init log failed\n");
return -1;
}

// Initialize the API client
ret = InitServiceAPI();
```

©2013-2022 Tencent Cloud. All rights reserved.

```
if ( ret != 0)
{
printf("init InitServiceAPI failed\n");
return -1;
}

// Send the request
ret = SendGetRequest();
if (0 != ret)
{
printf("SendGetRequest failed\n");
return -1;
}

// Receive the response
TcaplusServiceResponse* response = NULL;
ret = RecvResp(response);
if (0 != ret)
{
printf("RecvResp failed\n");
return -1;
}

// Get the result, where "0" indicates success
int32_t result = response->GetResult();
if (0 != result)
{
printf("the result is %d\n", result);
return -1;
}

// Get the record in the response
// Use "batch" command if there are multiple records in the response
int count = 0;
const TcaplusService::TcaplusServiceRecord* const_record = NULL;
while((count++) < response->GetRecordCount())
{
// Read the record
const_record = NULL;
printf("--- --- %3d --- ---\n", count - 1);
ret = response->FetchRecord(const_record);
if(0 != ret)
{
printf("FetchRecord() ret:%d\n", ret);
continue;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Assign the record to the structure
PLAYERONLINECNT stPLAYERONLINECNT;
memset(&stPLAYERONLINECNT, 0, sizeof(stPLAYERONLINECNT));
ret = const_record->GetData(&stPLAYERONLINECNT, sizeof(stPLAYERONLINECNT));
if(0 != ret)
{
printf("const_record->GetData() failed, ret: %d.", ret);
continue;
}

// Print the values in the structure
printf("struct dwTimeStamp %d szGameSvrID %s szGameAppID %s dwOnlineCntIOS %d dwO
nlineCntAndroid %d \n",
stPLAYERONLINECNT.dwTimeStamp,
stPLAYERONLINECNT.szGameSvrID,
stPLAYERONLINECNT.szGameAppID,
stPLAYERONLINECNT.dwOnlineCntIOS,
stPLAYERONLINECNT.dwOnlineCntAndroid);
}
return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Updating Data

Last updated：2021-01-04 14:35:23

Prerequisites

You have created a cluster, a table group and the `PLAYERONLINECNT` table.

The description file table_test.xml of the `PLAYERONLINECNT` table is as follows:

```
<?xml version="1.0" encoding="GBK" standalone="yes" ?>
<metalib name="tcaplus_tb" tagsetversion="1" version="1">
<struct name="PLAYERONLINECNT" version="1" primarykey="TimeStamp,GameSvrID" split
tablekey="TimeStamp">
<entry name="TimeStamp" type="uint32" desc="Measured in minutes" />
<entry name="GameSvrID" type="string" size="64" />
<entry name="GameAppID" type="string" size="64" desc="gameapp id" />
<entry name="OnlineCntIOS" type="uint32" defaultvalue="0" desc="The number of iOS
online users" />
<entry name="OnlineCntAndroid" type="uint32" defaultvalue="0" desc="The number of
Android online users" />
<entry name="BinaryLen" type="smalluint" defaultvalue="1" desc="Data length of th
e data source; skip the source check when the length is 0."/>
<entry name="binary" type="tinyint" desc="Binary" count= "1000" refer="BinaryLen"
/>
<entry name="binary2" type="tinyint" desc="Binary 2" count= "1000" refer="BinaryL
en" />
<entry name="strstr" type="string" size="64" desc="String"/>
<index name="index_id" column="TimeStamp"/>
</struct>
</metalib>
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.191.***.99:9999",
"tcp://10.191.***.88:9999"
};
// The number of target cluster addresses
```

©2013-2022 Tencent Cloud. All rights reserved.

```
static const int32_t DIR_URL_COUNT = 2;
// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "PLAYERONLINECNT" of the target business
static const char * TABLE_NAME = "PLAYERONLINECNT";
```

Step 2: Initialize TcaplusAPI log handles

Log configuration file: tlogconf.xml

```
// TCaplus service log class
TcaplusService::TLogger* g_pstTlogger;
LPTLOGCATEGORYINST g_pstLogHandler;
LPTLOGCTX g_pstLogCtx;
int32_t InitLog()
{
// Absolute path of the log configuration file
const char* sLogConfFile = "tlogconf.xml";
// Log class name
const char* sCategoryName = "mytest";
// Initialize the log handle using the configuration file
g_pstLogCtx = tlog_init_from_file(sLogConfFile);
if (NULL == g_pstLogCtx)
{
fprintf(stderr, "tlog_init_from_file failed.\n");
return -1;
}
// Get the log class
g_pstLogHandler = tlog_get_category(g_pstLogCtx, sCategoryName);
if (NULL == g_pstLogHandler)
{
fprintf(stderr, "tlog_get_category(mytest) failed.\n");
return -2;
}
// Initialize the log handle
g_pstTlogger = new TcaplusService::TLogger(g_pstLogHandler);
if (NULL == g_pstTlogger)
{
fprintf(stderr, "TcaplusService::TLogger failed.\n");
return -3;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}

return 0;
}
```

Step 3: Initialize TcaplusAPI client

```
// The client main class of the TCaplus service API
TcaplusService::TcaplusServer g_stTcapSvr;
// Meta information for the table
extern unsigned char g_szMetalib_tcaplus_tb[];
LPTDRMETA g_szTableMeta = NULL;
int32_t InitServiceAPI()
{
// Initialize
int32_t iRet = g_stTcapSvr.Init(g_pstTlogger, /*module_id*/0, /*app id*/APP_ID, /
*zone id*/ZONE_ID, /*signature*/SIGNATURE);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.Init failed, iRet: %d.", iRet);
return iRet;
}

// Add the directory server
for (int32_t i = 0; i< DIR_URL_COUNT; i++)
{
iRet = g_stTcapSvr.AddDirServerAddress(DIR_URL_ARRAY[i]);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.AddDirServerAddress(%s) failed, iR
et: %d.", DIR_URL_ARRAY[i], iRet);
return iRet;
}
}

// Get the meta description of the table
g_szTableMeta = tdr_get_meta_by_name((LPTDRMETALIB)g_szMetalib_tcaplus_tb, TABLE_
NAME);
if(NULL == g_szTableMeta)
{
tlog_error(g_pstLogHandler, 0, 0,"tdr_get_meta_by_name(%s) failed.", TABLE_NAME);
return -1;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Register the data table (connect to the directory server, verify the password,
and get the table routing) with 10-second timeout period
iRet = g_stTcapSvr.RegistTable(TABLE_NAME, g_szTableMeta, /*timeout_ms*/10000);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.RegistTable(%s) failed, iRet: %d."
, TABLE_NAME, iRet);
return iRet;
}

// Connect to all TcaplusDB proxy servers corresponding to the table
iRet = g_stTcapSvr.ConnectAll(/*timeout_ms*/10000, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.ConnectAll failed, iRet: %d.", iRe
t);
return iRet;
}

return 0;
}
```

Step 4. Send an update request via the API

```
int32_t SendUpdateRequest()
{
// Request object class
TcaplusService::TcaplusServiceRequest* pstRequest = g_stTcapSvr.GetRequest(TABLE_
NAME);
if (NULL == pstRequest)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.GetRequest(%s) failed.", TABLE_NAM
E);
return -1;
}

// Initialize the request object
int iRet = pstRequest->Init(TCAPLUS_API_UPDATE_REQ, NULL, 0, 0, 0, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->Init(TCAPLUS_API_UPDATE_REQ) faile
d, iRet: %d.", iRet);
return iRet;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Add a record to the table according to the request
TcaplusService::TcaplusServiceRecord* pstRecord = pstRequest->AddRecord();
if (NULL == pstRecord)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->AddRecord() failed.");
return -1;
}

PLAYERONLINECNT stPLAYERONLINECNT;
memset(&stPLAYERONLINECNT, 0, sizeof(stPLAYERONLINECNT));

// Specify the information of the key to be updated
stPLAYERONLINECNT.dwTimeStamp = 1;
snprintf(stPLAYERONLINECNT.szGameSvrID, sizeof(stPLAYERONLINECNT.szGameSvrID), "%
s", "mysvrid");

// Specify the information of the value to be updated
snprintf(stPLAYERONLINECNT.szGameAppID, sizeof(stPLAYERONLINECNT.szGameAppID), "%
s", "myappid2");
stPLAYERONLINECNT.dwOnlineCntIOS = 2;
stPLAYERONLINECNT.dwOnlineCntAndroid = 2;

// Set the record based on the TDR description
iRet = pstRecord->SetData(&stPLAYERONLINECNT, sizeof(stPLAYERONLINECNT));
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRecord->SetData() failed, iRet: %d.", iRet)
;
return iRet;
}

// Send the request packet
iRet= g_stTcapSvr.SendRequest(pstRequest);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.SendRequest failed, iRet: %d.", iR
et);
return iRet;
}
return 0;
}
```

Step 5. Receive the response via the API

©2013-2022 Tencent Cloud. All rights reserved.

```
int RecvResp(TcaplusServiceResponse*& response)
{
// Block the reception of response packets 5,000 times with each block lasting fo
r 1 ms
unsigned int sleep_us = 1000;
unsigned int sleep_count = 5000;
do
{
usleep(sleep_us);
response = NULL;
int ret = g_stTcapSvr.RecvResponse(response);
if (ret < 0)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse failed. ret:%d", r
et);
return ret;
}

// Receive a response packet
if (1 == ret)
{
break;
}
} while ((--sleep_count) > 0);

// Timeout period: 5 seconds
if (0 == sleep_count)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse wait timeout.");
return -1;
}
return 0;
}
```

Samples

Please refer to main.cpp.

```
int main(void) {
// Initialize the log
int ret = InitLog();
if ( ret != 0)
{
```

©2013-2022 Tencent Cloud. All rights reserved.

```
printf("init log failed\n");
return -1;
}

// Initialize the API client
ret = InitServiceAPI();
if ( ret != 0)
{
printf("init InitServiceAPI failed\n");
return -1;
}

// Send the request
ret = SendUpdateRequest();
if (0 != ret)
{
printf("SendUpdateRequest failed\n");
return -1;
}

// Receive the response
TcaplusServiceResponse* response = NULL;
ret = RecvResp(response);
if (0 != ret)
{
printf("RecvResp failed\n");
return -1;
}

// Get the result, where "0" indicates success
int32_t result = response->GetResult();
if (0 != result)
{
printf("the result is %d\n", result);
return -1;
}

return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

Deleting Data

Last updated：2021-01-04 14:35:23

Prerequisites

You have created a cluster, a table group and the `PLAYERONLINECNT` table.

The description file table_test.xml of the `PLAYERONLINECNT` table is as follows:

```
<?xml version="1.0" encoding="GBK" standalone="yes" ?>
<metalib name="tcaplus_tb" tagsetversion="1" version="1">
<struct name="PLAYERONLINECNT" version="1" primarykey="TimeStamp,GameSvrID" split
tablekey="TimeStamp">
<entry name="TimeStamp" type="uint32" desc="Measured in minutes" />
<entry name="GameSvrID" type="string" size="64" />
<entry name="GameAppID" type="string" size="64" desc="gameapp id" />
<entry name="OnlineCntIOS" type="uint32" defaultvalue="0" desc="The number of iOS
online users" />
<entry name="OnlineCntAndroid" type="uint32" defaultvalue="0" desc="The number of
Android online users" />
<entry name="BinaryLen" type="smalluint" defaultvalue="1" desc="Data length of th
e data source; skip the source check when the length is 0."/>
<entry name="binary" type="tinyint" desc="Binary" count= "1000" refer="BinaryLen"
/>
<entry name="binary2" type="tinyint" desc="Binary 2" count= "1000" refer="BinaryL
en" />
<entry name="strstr" type="string" size="64" desc="String"/>
<index name="index_id" column="TimeStamp"/>
</struct>
</metalib>
```

Step 1: Define configuration parameters

```
// Access address of the target cluster
static const char DIR_URL_ARRAY[][TCAPLUS_MAX_STRING_LENGTH] =
{
"tcp://10.191.***.99:9999",
"tcp://10.191.***.88:9999"
};
// The number of target cluster addresses
```

©2013-2022 Tencent Cloud. All rights reserved.

```
static const int32_t DIR_URL_COUNT = 2;
// Cluster ID of the target business
static const int32_t APP_ID = 3;
// Table group ID of the target business
static const int32_t ZONE_ID = 1;
// Target business password
static const char * SIGNATURE = "*******";
// Table name "PLAYERONLINECNT" of the target business
static const char * TABLE_NAME = "PLAYERONLINECNT";
```

Step 2: Initialize TcaplusAPI log handles

Log configuration file: tlogconf.xml

```
// TCaplus service log class
TcaplusService::TLogger* g_pstTlogger;
LPTLOGCATEGORYINST g_pstLogHandler;
LPTLOGCTX g_pstLogCtx;
int32_t InitLog()
{
// Absolute path of the log configuration file
const char* sLogConfFile = "tlogconf.xml";
// Log class name
const char* sCategoryName = "mytest";
// Initialize the log handle using the configuration file
g_pstLogCtx = tlog_init_from_file(sLogConfFile);
if (NULL == g_pstLogCtx)
{
fprintf(stderr, "tlog_init_from_file failed.\n");
return -1;
}
// Get the log class
g_pstLogHandler = tlog_get_category(g_pstLogCtx, sCategoryName);
if (NULL == g_pstLogHandler)
{
fprintf(stderr, "tlog_get_category(mytest) failed.\n");
return -2;
}
// Initialize the log handle
g_pstTlogger = new TcaplusService::TLogger(g_pstLogHandler);
if (NULL == g_pstTlogger)
{
fprintf(stderr, "TcaplusService::TLogger failed.\n");
return -3;
```

©2013-2022 Tencent Cloud. All rights reserved.

```
}

return 0;
}
```

Step 3: Initialize TcaplusAPI client

```
// The client main class of the TCaplus service API
TcaplusService::TcaplusServer g_stTcapSvr;
// Meta information for the table
extern unsigned char g_szMetalib_tcaplus_tb[];
LPTDRMETA g_szTableMeta = NULL;
int32_t InitServiceAPI()
{
// Initialize
int32_t iRet = g_stTcapSvr.Init(g_pstTlogger, /*module_id*/0, /*app id*/APP_ID, /
*zone id*/ZONE_ID, /*signature*/SIGNATURE);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.Init failed, iRet: %d.", iRet);
return iRet;
}

// Add the directory server
for (int32_t i = 0; i< DIR_URL_COUNT; i++)
{
iRet = g_stTcapSvr.AddDirServerAddress(DIR_URL_ARRAY[i]);
if (0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.AddDirServerAddress(%s) failed, iR
et: %d.", DIR_URL_ARRAY[i], iRet);
return iRet;
}
}

// Get the meta description of the table
g_szTableMeta = tdr_get_meta_by_name((LPTDRMETALIB)g_szMetalib_tcaplus_tb, TABLE_
NAME);
if(NULL == g_szTableMeta)
{
tlog_error(g_pstLogHandler, 0, 0,"tdr_get_meta_by_name(%s) failed.", TABLE_NAME);
return -1;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Register the data table (connect to the directory server, verify the password,
and get the table routing) with 10-second timeout period
iRet = g_stTcapSvr.RegistTable(TABLE_NAME, g_szTableMeta, /*timeout_ms*/10000);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.RegistTable(%s) failed, iRet: %d."
, TABLE_NAME, iRet);
return iRet;
}

// Connect to all TcaplusDB proxy servers corresponding to the table
iRet = g_stTcapSvr.ConnectAll(/*timeout_ms*/10000, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.ConnectAll failed, iRet: %d.", iRe
t);
return iRet;
}

return 0;
}
```

Step 4. Send a delete request via the API

```
int32_t SendDeleteRequest()
{
// Request object class
TcaplusService::TcaplusServiceRequest* pstRequest = g_stTcapSvr.GetRequest(TABLE_
NAME);
if (NULL == pstRequest)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.GetRequest(%s) failed.", TABLE_NAM
E);
return -1;
}

// Initialize the request object
int iRet = pstRequest->Init(TCAPLUS_API_DELETE_REQ, NULL, 0, 0, 0, 0);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->Init(TCAPLUS_API_DELETE_REQ) faile
d, iRet: %d.", iRet);
return iRet;
}
```

©2013-2022 Tencent Cloud. All rights reserved.

```
// Add a record to the table according to the request
TcaplusService::TcaplusServiceRecord* pstRecord = pstRequest->AddRecord();
if (NULL == pstRecord)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRequest->AddRecord() failed.");
return -1;
}

PLAYERONLINECNT stPLAYERONLINECNT;
memset(&stPLAYERONLINECNT, 0, sizeof(stPLAYERONLINECNT));

// Specify the information of the key to be deleted
stPLAYERONLINECNT.dwTimeStamp = 1;
snprintf(stPLAYERONLINECNT.szGameSvrID, sizeof(stPLAYERONLINECNT.szGameSvrID), "%
s", "mysvrid");


// Set the record based on the TDR description
iRet = pstRecord->SetData(&stPLAYERONLINECNT, sizeof(stPLAYERONLINECNT));
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "pstRecord->SetData() failed, iRet: %d.", iRet)
;
return iRet;
}

// Send the request packet
iRet= g_stTcapSvr.SendRequest(pstRequest);
if(0 != iRet)
{
tlog_error(g_pstLogHandler, 0, 0, "g_stTcapSvr.SendRequest failed, iRet: %d.", iR
et);
return iRet;
}
return 0;
}
```

Step 5. Receive the response via the API

```
int RecvResp(TcaplusServiceResponse*& response)
{
// Block the reception of response packets 5,000 times with each block lasting fo
r 1 ms
```

©2013-2022 Tencent Cloud. All rights reserved.

```
unsigned int sleep_us = 1000;
unsigned int sleep_count = 5000;
do
{
usleep(sleep_us);
response = NULL;
int ret = g_stTcapSvr.RecvResponse(response);
if (ret < 0)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse failed. ret:%d", r
et);
return ret;
}

// Receive a response packet
if (1 == ret)
{
break;
}
} while ((--sleep_count) > 0);

// Timeout period: 5 seconds
if (0 == sleep_count)
{
tlog_error(g_pstLogHandler, 0, 0, "tcaplus_server.RecvResponse wait timeout.");
return -1;
}
return 0;
}
```

Samples

Please refer to main.cpp.

```
int main(void) {
// Initialize the log
int ret = InitLog();
if ( ret != 0)
{
printf("init log failed\n");
return -1;
}

// Initialize the API client
```

©2013-2022 Tencent Cloud. All rights reserved.

```
ret = InitServiceAPI();
if ( ret != 0)
{
printf("init InitServiceAPI failed\n");
return -1;
}

// Send the request
ret = SendDeleteRequest();
if (0 != ret)
{
printf("SendDeleteRequest failed\n");
return -1;
}

// Receive the response
TcaplusServiceResponse* response = NULL;
ret = RecvResp(response);
if (0 != ret)
{
printf("RecvResp failed\n");
return -1;
}

// Get the result, where "0" indicates success
int32_t result = response->GetResult();
if (0 != result)
{
printf("the result is %d\n", result);
return -1;
}

return 0;
}
```

©2013-2022 Tencent Cloud. All rights reserved.