Mobile Live Video Broadcasting

Trial Demo

Product Documentation

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

Trial Demo

Trial Demo

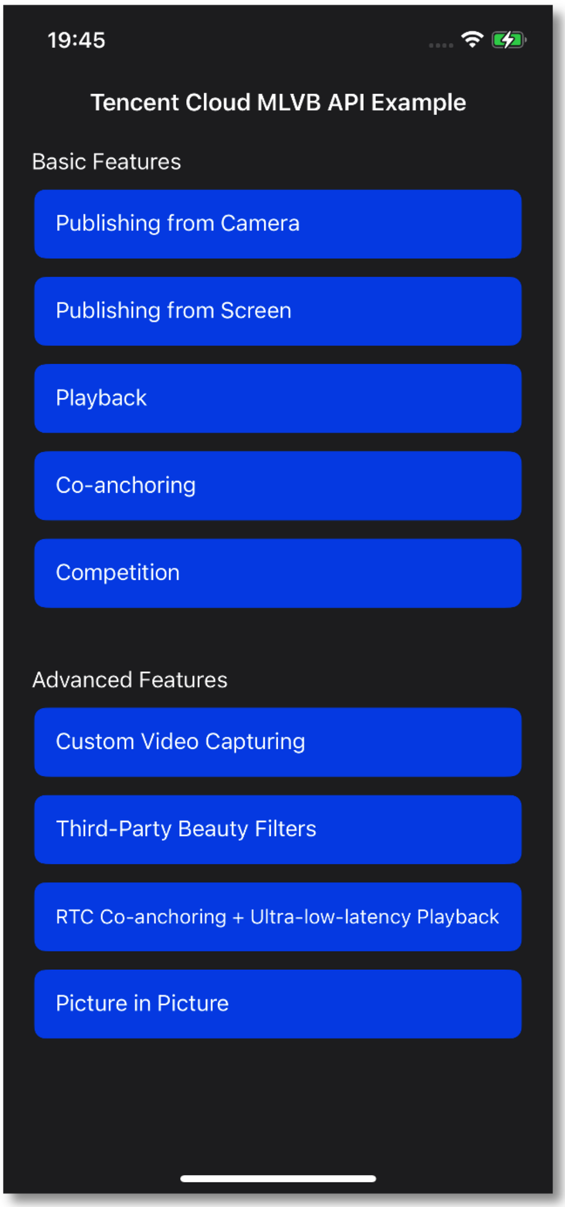
Last updated：2022-10-20 15:16:48

API Samples for Development and Debugging

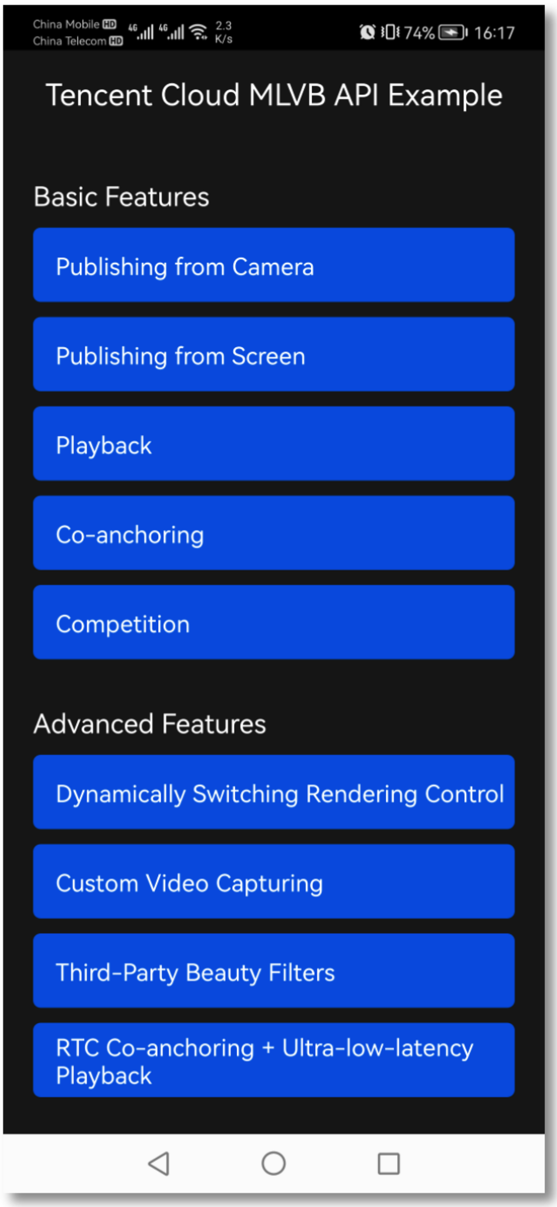
`MLVB API-Example Demo` is provided to help you better understand MLVB SDK APIs to quickly implement the basic features of some live streaming scenarios. You can quickly run this demo as instructed in the following documents:

Platform	Source Code Address	How to Run the Demo
Android	MLVB API-Example	Android
iOS	MLVB-API-Example-OC	iOS

iOS


19:45
Tencent Cloud MLVB API Example
Basic Features
Publishing from Camera
Publishing from Screen
Playback
Co-anchoring
Competition
Advanced Features
Custom Video Capturing
Third-Party Beauty Filters
RTC Co-anchoring + Ultra-low-latency Playback
Picture in Picture

Android


China Mobile
China Telecom
74%
16:17
Tencent Cloud MLVB API Example
Basic Features
Publishing from Camera
Publishing from Screen
Playback
Co-anchoring
Competition
Advanced Features
Dynamically Switching Rendering Control
Custom Video Capturing
Third-Party Beauty Filters
RTC Co-anchoring + Ultra-low-latency Playback

Running a Demo
iOS

Last updated：2022-10-20 15:15:10

This document describes how to quickly run Tencent Cloud MLVB-API-Example for iOS.

Environment Requirements

- Xcode 9.0 or later
- iPhone or iPad with iOS 9.0 or later
- A valid developer signature for your project

Prerequisites

You have signed up for a Tencent Cloud account.

Directions

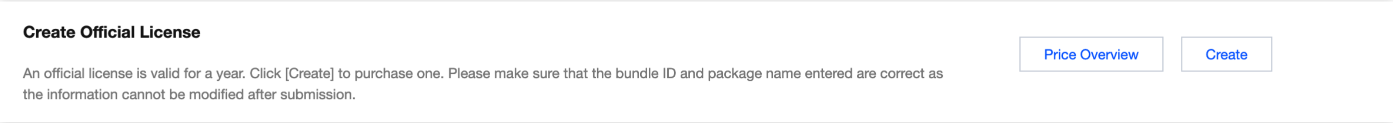
Step 1. Download the SDK and MLVB-API-Example source code

1. Download the package here as needed. Here, the Professional Edition is used as an example.
2. Decompress the file after download.

> Note：
>
> The source code can also be obtained from GitHub.

Step 2. Configure the license

1. Log in to the CSS console, select **MLVB SDK** > **License Management** on the left sidebar, and click **Create**.



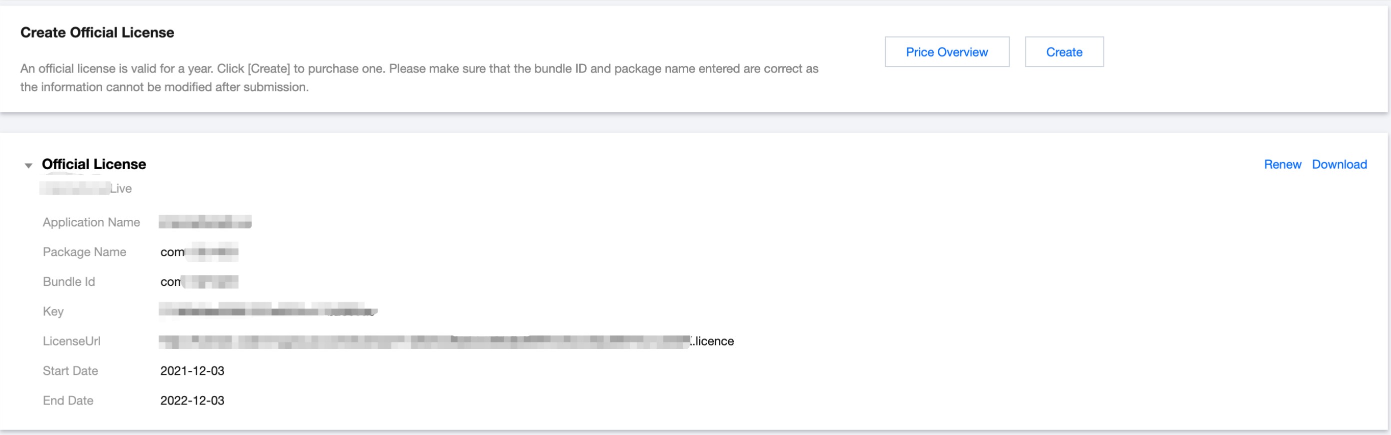
Create Official License

An official license is valid for a year. Click [Create] to purchase one. Please make sure that the bundle ID and package name entered are correct as the information cannot be modified after submission.

2. Enter the `App Name`, `Package Name`, and `Bundle ID` as needed, select the **Live streaming** feature module (**Live Push** + **Video Playback**), and click **Confirm**.

- **Package Name**: Enter the **applicationId** in the **build.gradle** file in the `App` directory.
- **Bundle ID**: Enter the **Bundle Identifier** of the project in **Xcode**.

3. After the free trial license is created successfully, the page will display the information of the generated license. **You need to pass in two parameters `Key` and `License URL` during initial SDK configuration. Store the following information properly:**



Create Official License

An official license is valid for a year. Click [Create] to purchase one. Please make sure that the bundle ID and package name entered are correct as the information cannot be modified after submission.

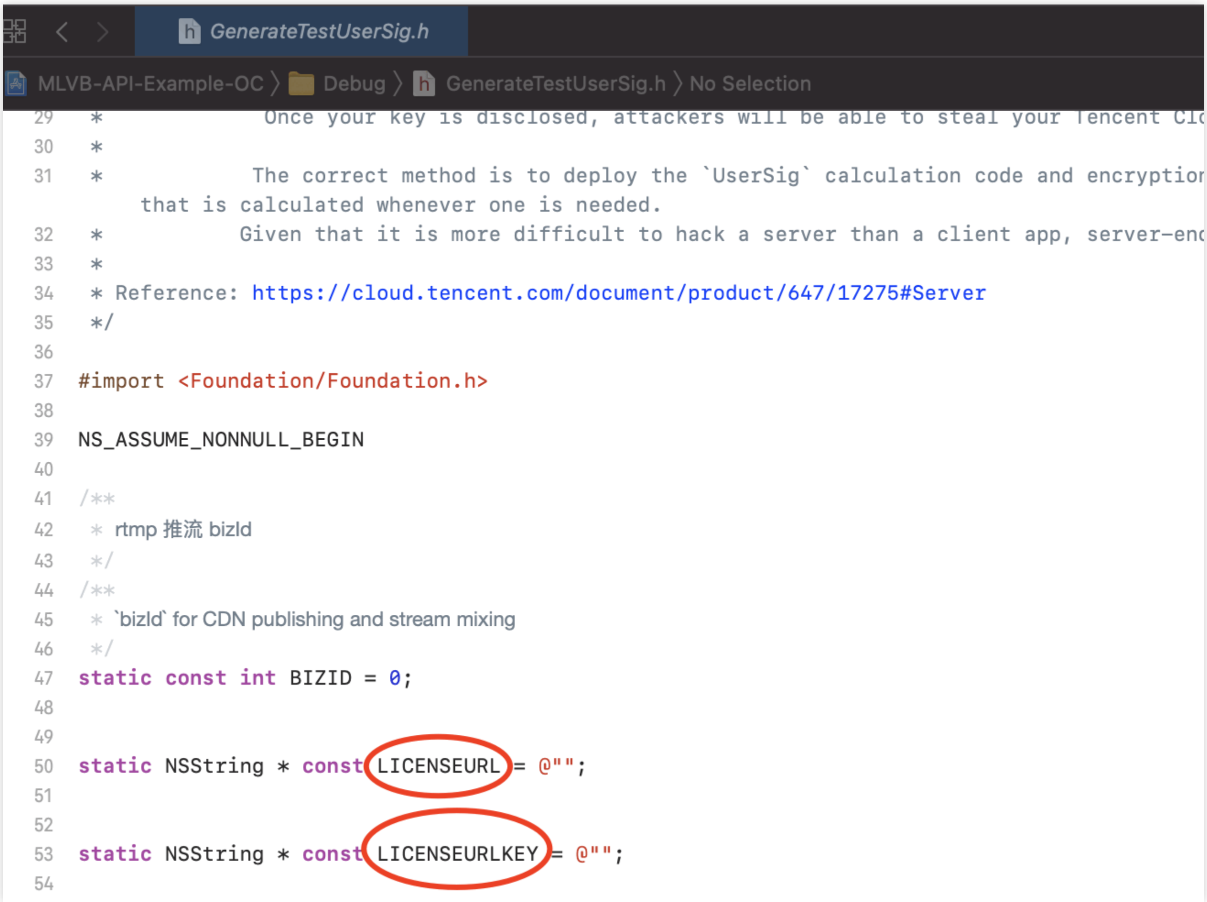
Official License

[redacted]Live

Application Name	[redacted]
Package Name	com[redacted]
Bundle Id	con[redacted]
Key	[redacted]
LicenseUrl	[redacted].licence
Start Date	2021-12-03
End Date	2022-12-03

4. Open the `LiteAVSDK_Professional_iOS_version number/MLVB-API-Example-OC/Debug/GenerateTestUserSig.h` file.
Set parameters in `GenerateTestUserSig.h` as follows:
   - LICENSEURL: Empty by default. Set it to the actual download license URL.

- LICENSEURLKEY: Empty by default. Set it to the actual download license key.

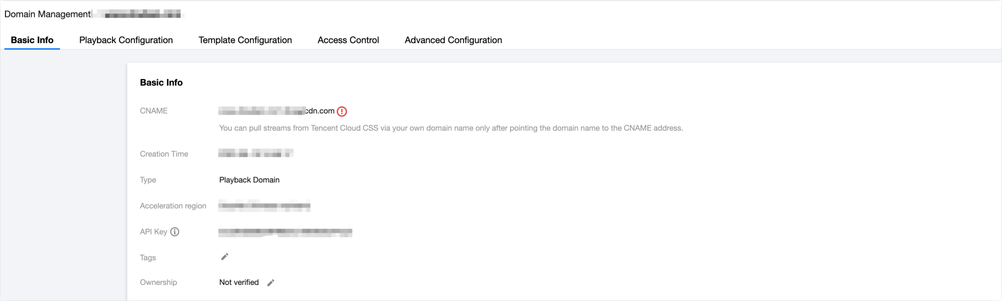


GenerateTestUserSig.h
MLVB-API-Example-OC 〉 Debug 〉 GenerateTestUserSig.h 〉 No Selection

```
 *              Once your key is disclosed, attackers will be able to steal your Tencent Clo
 *
 *             The correct method is to deploy the `UserSig` calculation code and encryption
     that is calculated whenever one is needed.
 *           Given that it is more difficult to hack a server than a client app, server-end
 *
 * Reference: https://cloud.tencent.com/document/product/647/17275#Server
 */

#import <Foundation/Foundation.h>

NS_ASSUME_NONNULL_BEGIN

/**
 * rtmp 推流 bizId
 */
/**
 * `bizId` for CDN publishing and stream mixing
 */
static const int BIZID = 0;


static NSString * const LICENSEURL = @"";


static NSString * const LICENSEURLKEY = @"";

```

Step 3. Configure stream push/playback capabilities

1. Apply for a domain name in DNSPod and get an ICP filing for it.
2. Add the stream push/playback domain name in **Domain Management** in the **CSS console**. For detailed directions, see Adding Your Own Domain.
3. Configure the CNAME record for the domain name as instructed in Configuring CNAME.
4. After configuring the stream push/playback domain name, you can get the `CNAME` information on the **Basic Info** page of the domain name.



Basic Info

CNAME [redacted]cdn.com

You can pull streams from Tencent Cloud CSS via your own domain name only after pointing the domain name to the CNAME address.

Creation Time [redacted]

Type Playback Domain

Acceleration region [redacted]

API Key [redacted]

Tags

Ownership Not verified

5. Open the `LiteAVSDK_Professional_iOS_version number/MLVB-API-Example-OC/Debug/GenerateTestUserSig.h` file.
   Set parameters in `GenerateTestUserSig.h` as follows:

- **PUSH_DOMAIN**: Set it to your stream push domain name.
- **PLAY_DOMAIN**: Set it to your playback domain name.
- **LIVE_URL_KEY**: This parameter is optional. It is used to generate authentication information such as `txSecret` . For more information on how to calculate it, see Publishing/Playback URL. You can query it in **Manage** > **Stream Push Configuration** > **Authentication Configuration** on the Domain Name page.

Configuring stream push parameters

1. Find and open the `LiteAVSDK_Professional_iOS_version number/MLVB-API-Example-OC/Debug/GenerateTestUserSig.h` file.
2. Set parameters in the GenerateTestUserSig.h file based on the above service:

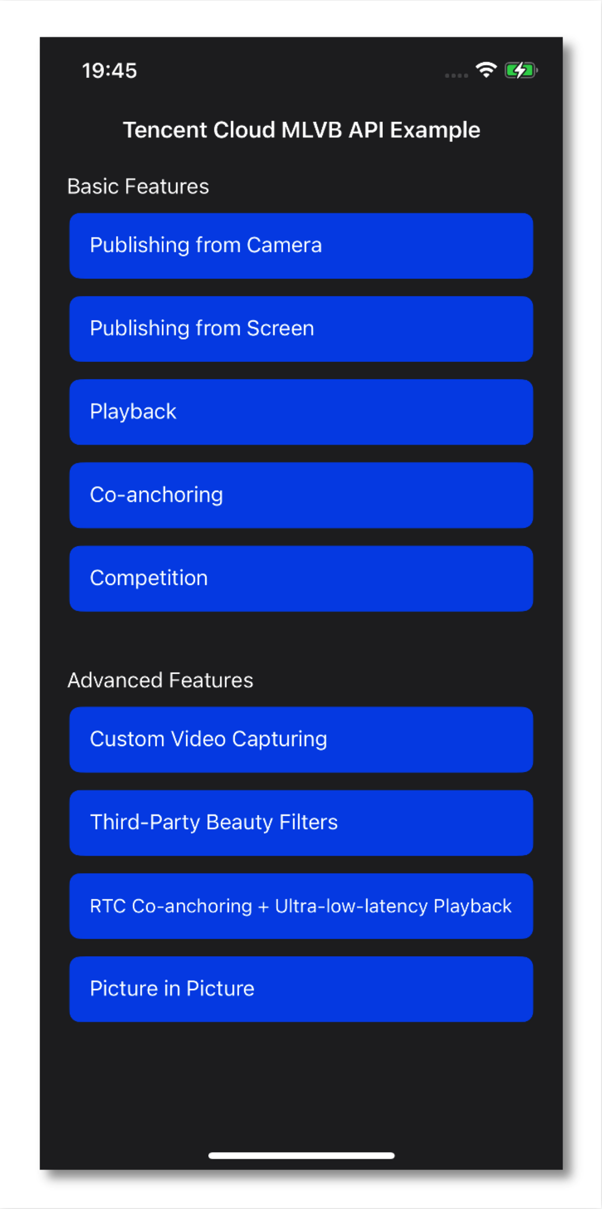
- SDKAppID: `0` by default. Set it to the actual `SDKAppID` .
- SECRETKEY: Empty by default. Set it to the actual secret key.

Stream push URL field description

You need to concatenate the specific stream push/pull URL string based on the used protocol as instructed in Publishing/Playback URL. A string has been concatenated in the demo, and the stream can be played back after you run the demo.

Step 5. Compile and run

Open the demo project `MLVB-API-Example-OC` with Xcode 9.0 or later and click **Run**.



Android

Last updated：2022-10-20 15:15:10

This document describes how to quickly run the Tencent Cloud MLVB-API-Example for Android.

Environment Requirements

- Android 4.1 (SDK API level 16) or above; Android 5.0 (SDK API level 21) or above is recommended.
- Android Studio 3.5 or later
- Device on Android 4.1 or above for the application

Prerequisites

You have signed up for a Tencent Cloud account.

Directions

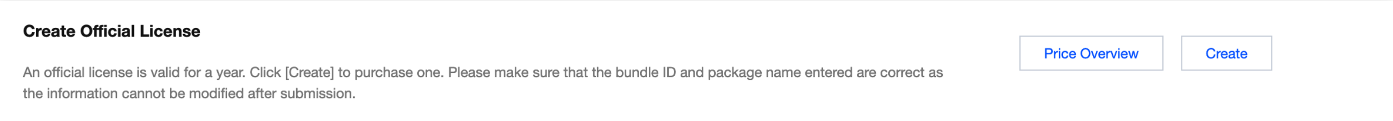
Step 1. Download the SDK and MLVB-API-Example source code

1. Download the package here as needed. Here, the Professional Edition is used as an example.
2. Decompress the file after download.

> Note：
>
> The source code can also be obtained from GitHub.

Step 2. Configure the license

1. Log in to the CSS console, select **MLVB SDK** > **License Management** on the left sidebar, and click **Create**.



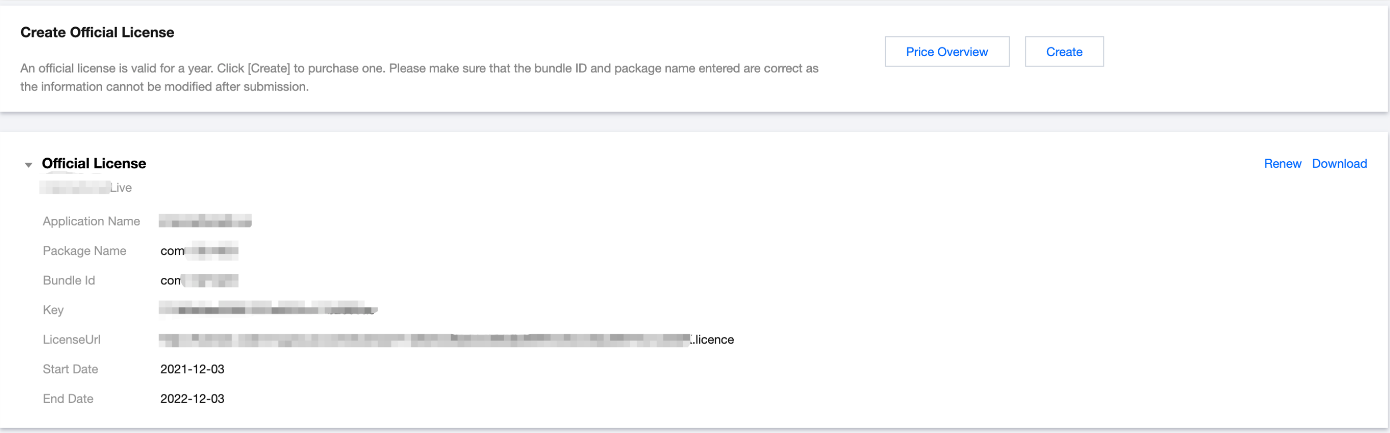
Create Official License

An official license is valid for a year. Click [Create] to purchase one. Please make sure that the bundle ID and package name entered are correct as the information cannot be modified after submission.

2. Enter the `App Name` , `Package Name` , and `Bundle ID` as needed and click **Confirm**.

- **Package Name**: Enter the **applicationId** in the **build.gradle** file in the `App` directory.
- **Bundle ID**: Enter the **Bundle Identifier** of the project in **Xcode**.

3. After the license is created successfully, the page will display the information of the generated license. **You need to pass in two parameters, `Key` and `License URL` , during initial SDK configuration. Store the following information properly.**



Create Official License

An official license is valid for a year. Click [Create] to purchase one. Please make sure that the bundle ID and package name entered are correct as the information cannot be modified after submission.

Official License

[redacted]Live

Application Name [redacted]

Package Name com[redacted]

Bundle Id con[redacted]

Key [redacted]

LicenseUrl [redacted].licence

Start Date 2021-12-03

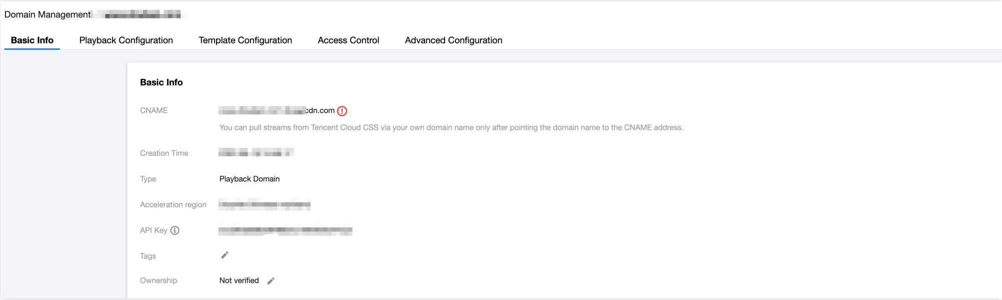
End Date 2022-12-03

4. Open the `LiteAVSDK_Professional_Android_version number/MLVB-API-Example/Debug/src/main/java/com/tencent/mlvb/debug/GenerateTestUserSig.java` file. Set parameters in `GenerateTestUserSig.java` as follows:

- `LICENSEURL` : A placeholder by default. Set it to the actual download license URL.
- LICENSEURLKEY: A placeholder by default. Set it to the actual download license key.

Step 3. Configure stream push/playback capabilities

1. Apply for a domain name in DNSPod and get an ICP filing for it.
2. Add the stream push/playback domain name in **Domain Management** in the **CSS console**. For detailed directions, see Adding Your Own Domain.
3. Configure the CNAME record for the domain name as instructed in Configuring CNAME.
4. After configuring the stream push/playback domain name, you can get the `CNAME` information on the **Basic Info** page of the domain name.



Domain Management[redacted]

Basic Info

CNAME [redacted]cdn.com ⓘ

You can pull streams from Tencent Cloud CSS via your own domain name only after pointing the domain name to the CNAME address.

Creation Time [redacted]

Type Playback Domain

Acceleration region [redacted]

API Key ⓘ [redacted]

Tags

Ownership Not verified

5. Open the `LiteAVSDK_Professional_Android_version number/MLVB-API-Example/Debug/src/main/java/com/tencent/mlvb/debug/GenerateTestUserSig.java` file. Set parameters in `GenerateTestUserSig.java` as follows:

- **PUSH_DOMAIN**: Set it to your stream push domain name.
- **PLAY_DOMAIN**: Set it to your playback domain name.
- **LIVE_URL_KEY**: This parameter is optional. It is used to generate authentication information such as `txSecret` . For more information on how to calculate it, see Publishing/Playback URL. You can query it in **Manage** > **Stream Push Configuration** > **Authentication Configuration** on the Domain Name page.

Configuring stream push parameters

1. Find and open the `LiteAVSDK_Professional_Android version number/MLVB-API-Example/Debug/src/main/java/com/tencent/mlvb/debug/GenerateTestUserSig.java` file.
2. Set parameters in the GenerateTestUserSig.java file based on the above service:

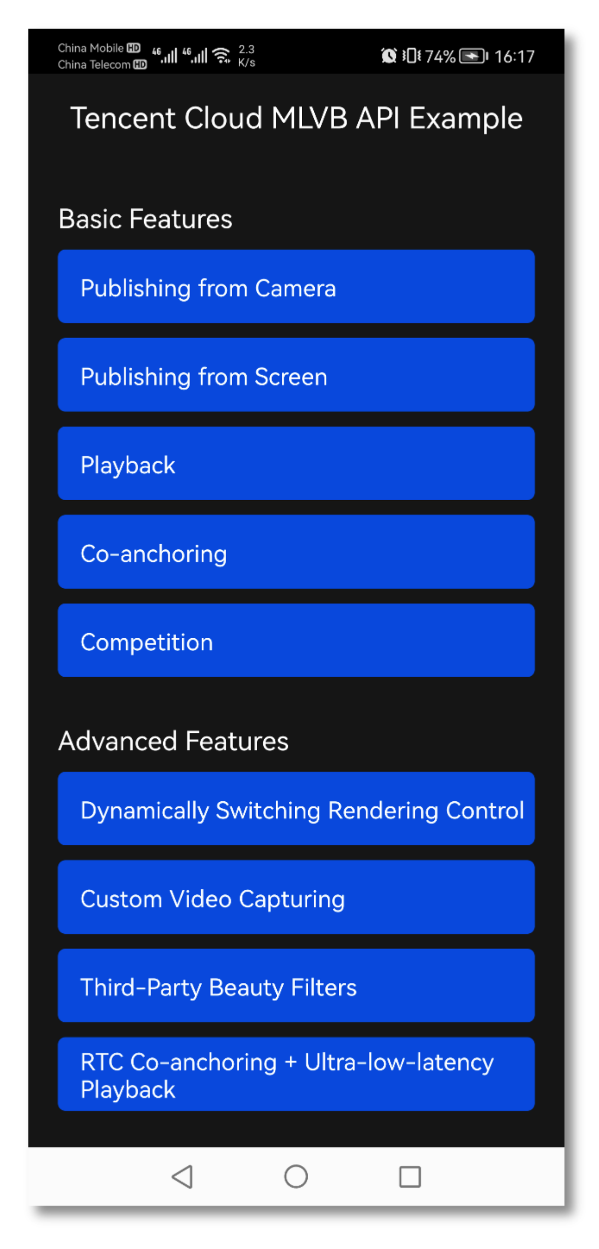
- SDKAPPID: A placeholder by default. Set it to the actual `SDKAppID` .
- SECRETKEY: A placeholder by default. Set it to the actual secret key.

Stream push URL field description

You need to concatenate the specific stream push/pull URL string based on the used protocol as instructed in Publishing/Playback URL. A string has been concatenated in the demo, and the stream can be played back after you run the demo.

Step 5. Compile and run

Open the demo project `MLVB-API-Example` with Android Studio 3.5 or later and click **Run**.



China Mobile HD
China Telecom HD
2.3 K/s
74%
16:17
Tencent Cloud MLVB API Example
Basic Features
Publishing from Camera
Publishing from Screen
Playback
Co-anchoring
Competition
Advanced Features
Dynamically Switching Rendering Control
Custom Video Capturing
Third-Party Beauty Filters
RTC Co-anchoring + Ultra-low-latency Playback