

移动直播 SDK

直播播放

产品文档



腾讯云

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

直播播放

iOS

标准直播拉流

快直播拉流

Android

标准直播拉流

快直播拉流

直播播放

iOS

标准直播拉流

最近更新时间：2022-06-14 13:00:34

基础知识

本文主要介绍视频云 SDK 的直播播放功能。

直播和点播

- **直播 (LIVE)** 的视频源是主播实时推送的。因此，主播停止推送后，播放端的画面也会随即停止，而且由于是实时直播，所以播放器在播直播 URL 的时候是没有进度条的。
- **点播 (VOD)** 的视频源是云端的一个视频文件，只要未被从云端移除，视频就可以随时播放，播放中您可以通过进度条控制播放位置，腾讯视频和优酷土豆等视频网站上的视频观看就是典型的点播场景。

协议的支持

通常使用的直播协议如下，标准直播推荐使用 FLV 协议的直播地址（以 `http` 开头，以 `.flv` 结尾），快直播使用 WebRTC 协议，更多信息请参见：[快直播拉流](#)

直播协议	优点	缺点	播放延迟
FLV	成熟度高、高并发无压力	需集成 SDK 才能播放	2s - 3s
RTMP	延迟较低	高并发情况下表现不佳	1s - 3s
HLS(m3u8)	手机浏览器支持度高	延迟非常高	10s - 30s
WebRTC	延迟最低	需集成 SDK 才能播放	< 1s

说明：

标准直播与快直播计费价格不同，更多计费详情请参见 [标准直播计费](#) 和 [快直播计费](#)。

特别说明

视频云 SDK 不会对播放地址的来源做限制，即您可以用它来播放腾讯云或非腾讯云的播放地址。但视频云 SDK 中的播放器只支持 FLV、RTMP、HLS（m3u8）和 WebRTC 四种格式的直播地址，以及 MP4、HLS（m3u8）和 FLV 三种格式的点播地址。

示例代码

针对开发者的接入反馈的高频问题，腾讯云提供有更加简洁的 API-Example 工程，方便开发者可以快速的了解相关 API 的使用，欢迎使用。

所属平台	GitHub 地址
iOS	Github
Android	Github

对接攻略

步骤1：创建 Player

视频云 SDK 中的 V2TXLivePlayer 模块负责实现直播播放功能。

```
V2TXLivePlayer *_txLivePlayer = [[V2TXLivePlayer alloc] init];
```

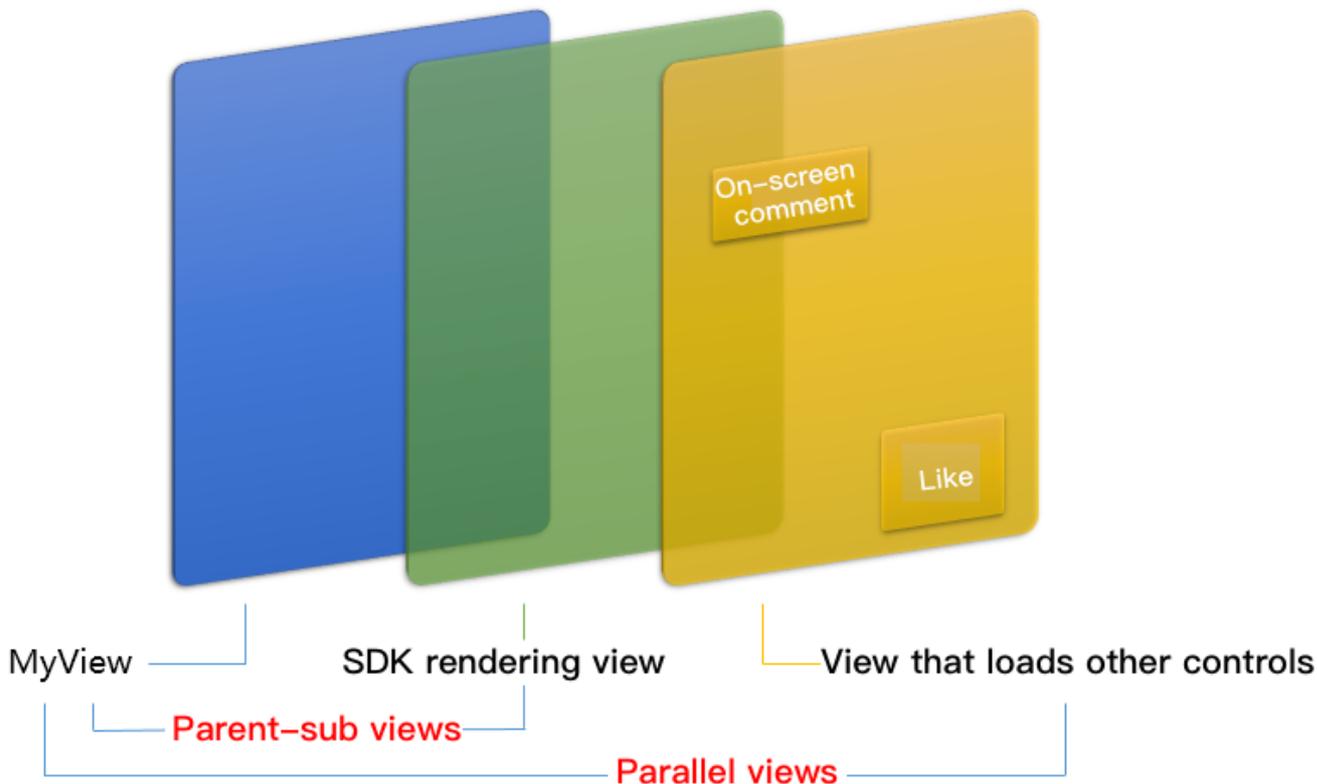
步骤2：渲染 View

接下来我们要给播放器的视频画面找个地方来显示，iOS 系统中使用 **view** 作为基本的界面渲染单位，所以您只需要准备一个 **view** 并调整好布局就可以了。

```
//用 setRenderView 给播放器绑定决定渲染区域的view。  
[_txLivePlayer setRenderView:_myView];
```

内部原理上，播放器并不是直接把画面渲染到您提供的 **view**（示例代码中的 `_myView`）上，而是在这个 **view** 之上创建一个用于 OpenGL 渲染的子视图（**subView**）。

如果您要调整渲染画面的大小，只需要调整您所常见的 **view** 的大小和位置即可，SDK 会让视频画面跟着您的 **view** 的大小和位置进行实时的调整。



如何做动画？

针对 view 做动画是比较自由的，不过请注意此处动画所修改的目标属性应该是 transform 属性而不是 frame 属性。

```
[UIView animateWithDuration:0.5 animations:^(
    _myView.transform = CGAffineTransformMakeScale(0.3, 0.3); //缩小1/3
)];
```

步骤3：启动播放

```
NSString* url = @"http://2157.liveplay.myqcloud.com/live/2157_xxxx.flv";
[_txLivePlayer startPlay:url];
```

步骤4：画面调整

- **setRenderFillMode**：铺满 or 适应

可选值	含义
V2TXLiveFillModeFill	将图像等比例铺满整个屏幕，多余部分裁剪掉，此模式下画面不会留黑边，但可能因为部分区域被裁剪而显示不全
V2TXLiveFillModeFit	将图像等比例缩放，适配最长边，缩放后的宽和高都不会超过显示区域，居中显示，画面可能会留有黑边

• **setRenderRotation**：视频画面顺时针旋转角度

可选值	含义
V2TXLiveRotation0	不旋转
V2TXLiveRotation90	顺时针旋转90度
V2TXLiveRotation180	顺时针旋转180度
V2TXLiveRotation270	顺时针旋转270度



The long side fits the screen



The short side fits the screen



Landscape

步骤5：暂停播放

对于直播播放而言，并没有真正意义上的暂停，所谓的直播暂停，只是**画面冻结**和**关闭声音**，而云端的视频源还在不断地更新着，所以当您调用 `resume` 的时候，会从最新的时间点开始播放，这是和点播对比的最大不同点（点播播放器的暂停和继续与播放本地视频文件时的表现相同）。

```
// 暂停
[_txLivePlayer pauseAudio];
[_txLivePlayer pauseVideo];
// 恢复
```

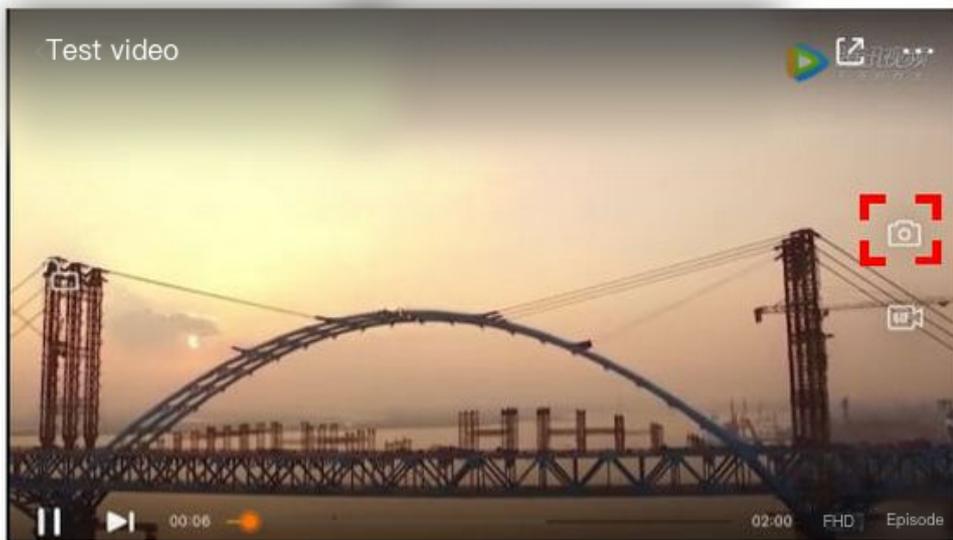
```
[_txLivePlayer resumeAudio];
[_txLivePlayer resumeVideo];
```

步骤6：结束播放

```
// 停止播放
[_txLivePlayer stopPlay];
```

步骤7：屏幕截图

通过调用 **snapshot** 您可以截取当前直播画面为一帧屏幕通过 V2TXLivePlayerObserver 的 **onSnapshotComplete** 回调截屏图片，此功能只会截取当前直播流的视频画面，如果您需要截取当前的整个 UI 界面，请调用 iOS 的系统 API 来实现。



➔ Screenshot
Video (not UI) screenshot

```
...
[_txLivePlayer setObserver:self];
[_txLivePlayer snapshot];
...
- (void)onSnapshotComplete:(id<V2TXLivePlayer>)player image:(TXImage *)image {
    if (image != nil) {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self handle:image];
        });
    }
}
```

延时调节

腾讯云 SDK 的直播播放功能，并非基于 `ffmpeg` 做二次开发，而是采用了自研的播放引擎，所以相比于开源播放器，在直播的延迟控制方面有更好的表现，我们提供了三种延迟调节模式，分别适用于：秀场，游戏以及混合场景。

• 三种模式的特性对比

控制模式	卡顿率	平均延迟	适用场景	原理简述
极速模式	较流畅 偏高	2s- 3s	美女秀场（冲顶大会）	在延迟控制上有优势，适用于对延迟大小比较敏感的场景
流畅模式	卡顿率 最低	>= 5s	游戏直播（企鹅电竞）	对于超大码率的游戏直播（例如绝地求生）非常适合，卡顿率最低
自动模式	网络自 适应	2s-8s	混合场景	观众端的网络越好，延迟就越低；观众端网络越差，延迟就越高

• 三种模式的对接代码

```
//自动模式
[_txLivePlayer setCacheParams:1 maxTime:5];
//极速模式
[_txLivePlayer setCacheParams:1 maxTime:1];
//流畅模式
[_txLivePlayer setCacheParams:5 maxTime:5];
//设置完成之后再启动播放
```

说明：

更多关于卡顿和延迟优化的技术知识，请参见 [如何优化视频卡顿？](#)

SDK 事件监听

您可以为 `V2TXLivePlayer` 对象绑定一个 `V2TXLivePlayerObserver`，之后 SDK 的内部状态信息例如播放器状态、播放音量回调、音视频首帧回调、统计数据、警告和错误信息等会通过对应的回调通知给您。

定时触发的状态通知

- **onStatisticsUpdate** 通知每2秒都会被触发一次，目的是实时反馈当前的播放器状态，它就像汽车的仪表盘，可以告知您目前 SDK 内部的一些具体情况，以便您能对当前网络状况和视频信息等有所了解。

评估参数	含义说明
appCpu	当前 App 的 CPU 使用率 (%)
systemCpu	当前系统的 CPU 使用率 (%)
width	视频宽度
height	视频高度
fps	帧率 (fps)
audioBitrate	音频码率 (Kbps)
videoBitrate	视频码率 (Kbps)

- **onPlayoutVolumeUpdate** 播放器音量大小回调。这个回调仅当您调用 **enableVolumeEvaluation** 开启播放音量大小提示之后才会工作。回调的时间间隔也会与您在设置 `enableVolumeEvaluation` 的参数 `intervalMs` 保持一致。

非定时触发的状态通知

其余的回调仅在事件发生时才会抛出来。

快直播拉流

最近更新时间：2022-06-14 13:01:03

快直播概述

快直播（Live Event Broadcasting, LEB）是标准直播在超低延迟播放场景下的延伸，比传统直播协议延迟更低，为观众提供毫秒级的极致直播观看体验。能够满足一些对延迟性能要求更高的特定场景需求，例如在线教育、体育赛事直播、在线答题等。

说明：

- 上图为快直播和标准的 CDN 直播的真实对比视频（使用 [screpy 工具](#) 配合录制），从左至右分别为：标准的 CDN 直播、**快直播**、推流端。
- 标准直播与快直播计费价格不同，更多计费详情请参见 [标准直播计费](#) 和 [快直播计费](#)。

方案优势

优势	说明
毫秒级超低延迟播放	采用 UDP 协议将传统直播中3秒 - 5秒延迟降低至1秒以内，同时兼顾秒开、卡顿率等核心指标，给用户带来极致的超低延迟直播体验。
功能完善，平滑兼容	兼容了标准直播包括推流、转码、录制、截图、鉴黄、播放等全功能，支持客户从现有的标准直播业务平滑迁移。
简单易用，安全可靠	采用标准协议，对接简单，在 Chrome 和 Safari 浏览器中无需任何插件即可进行播放。播放协议默认加密，更加安全可靠。

适用场景

场景	说明
体育赛事	快直播为体育赛事提供超低延迟的直播能力加持，使比赛赛事结果快速通过直播触达用户，让观众享受实时了解赛事动态的乐趣。
电商直播	电商直播中，商品拍卖、促销抢购等交易反馈对直播实时性要求很高，快直播的超低延迟能力，能让主播和观众能够及时得到交易反馈，提升边看边买的体验。
在线课堂	师生通过直播完成在线的课堂教学，得益于快直播的超低延迟能力，使课堂互动能力得到提升，让在线课堂也能像线下授课一样自然。

场景	说明
在线答题	传统的在线答题由于存在延迟，观众端有时需要进行补帧才能让观众主持两端同时显示。快直播的超低延迟能够完美解决这个问题，让双方实时看到答题画面，降低了实现难度，也让体验更加流畅。
秀场互动	快直播适用于秀场直播场景，极大优化了在观众送礼等对画面实时性要求高的直播互动场景中的观众互动体验。

体验快直播

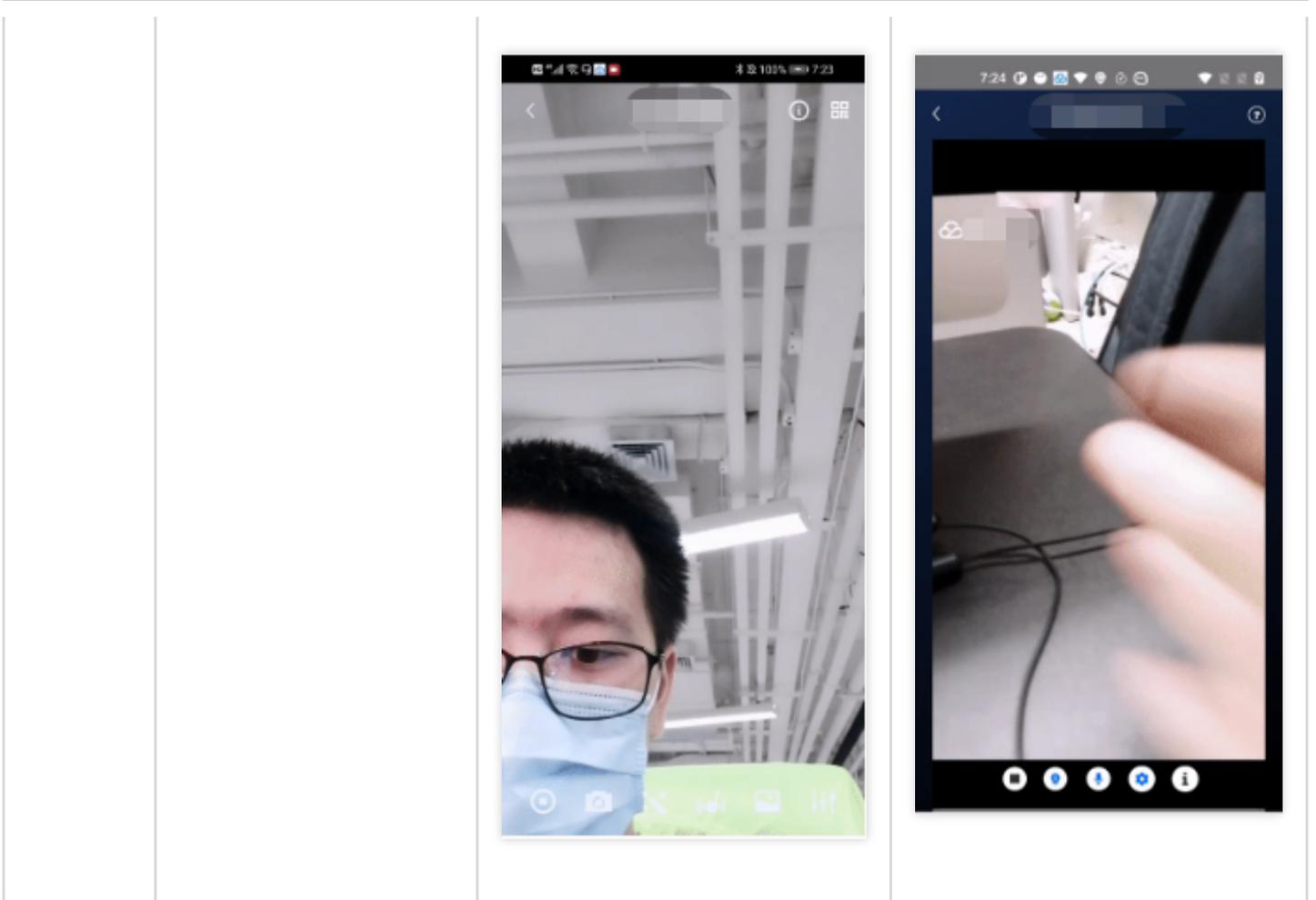
视频云工具包是腾讯云开源的一套完整的音视频服务解决方案，包含实时音视频（TRTC）、移动直播（MLVB）、短视频（UGC）等多个 SDK 的能力展示，其中包含快直播相关体验 UI — **快直播播放**。

说明：

Demo 演示和体验步骤以 Android 为例，iOS Demo 界面略有不同。

源码及示例

源码下载	体验安装	推流演示 (Android)	播放演示 (Android)
Android			
iOS			



说明：

除上述示例外，针对开发者的接入反馈的高频问题，腾讯云提供更加简洁的 API-Example 工程，方便开发者可以快速的了解相关 API 的使用，欢迎使用。

- iOS：[MLVB-API-Example](#)
- Android：[MLVB-API-Example](#)

推流体验

快直播兼容了标准直播，因此可以使用普通推流端进行推流，然后使用快直播进行拉流。

1. 下载 [视频云工具包](#)，安装登录后，进入【推流演示（摄像头推流）】中。
2. 允许相关权限申请，单击【自动生成推流地址】，即开始推流了。
3. 成功推流之后，点击右上角的二维码图标，可以获取播放地址，其中【快直播】就是用于快直播的播放地址。
4. 成功开始推流后，可单击右下侧的菜单按钮，进行美颜、BGM、切换摄像头等设置操作。

播放体验

1. 下载 [视频云工具包](#)，安装登录后，进入【快直播播放】中。

2. 允许相关权限申请，单击二维码扫描按钮，扫描【推流体验】中得到的快直播播放地址。
3. 扫描完成后即开始播放，播放成功后，可单击右下侧的菜单按钮，进行静音、设置等操作。

接入工程

新版本的移动直播 SDK，可以使用 `V2TXLivePlayer` 来播放快直播的流，同时也提供了 `V2TXLivePusher` 来推流。快直播直播协议支持 WebRTC 标准协议，使用标准的扩展方式，其 URL 均以 `webrtc://` 字符开始。

步骤1：下载 SDK

可以在 [SDK 下载](#) 页面选择全功能版或者企业版下载。

步骤2：获取播放 URL

在直播场景中，不论是推流还是拉流都离不开对应的 URL。请参见 [快直播快速入门](#) 获取快直播的播放 URL。快直播 URL 均以 `webrtc://` 字符开始，类似于这样：

```
webrtc://{Domain}/{AppName}/{StreamName}
```

在上述的 URL 中，存在一些关键字段，关于其中关键字段的含义信息，详见下表：

字段名称	字段含义
webrtc://	快直播 URL 的前缀字段
Domain	快直播播放域名
AppName	应用名称，指的是直播流媒体文件存放路径，默认云直播会分配一个路径：live
StreamName	流名称，指每路直播流唯一的标识符

说明：

如果需要推流，具体操作请参见 [摄像头推流](#) 或者 [录屏推流](#)。

步骤3：实现快直播播放

使用 `V2TXLivePlayer` 对象可以使用快直播进行拉流，具体做法如下（传入正确的 URL 是关键）：

示例代码

- [Android](#)
- [iOS](#)

```
// 创建一个 V2TXLivePlayer 对象；
V2TXLivePlayer player = new V2TXLivePlayerImpl(mContext);
player.setObserver(new MyPlayerObserver(playerView));
player.setRenderView(mSurfaceView);
// 传入低延时协议播放地址，即可开始播放；
player.startPlay("webrtc://{Domain}/{AppName}/{StreamName}");
```

Android

标准直播拉流

最近更新时间：2022-06-14 13:01:45

基础知识

本文主要介绍视频云 SDK 的直播播放功能。

直播和点播

- **直播 (LIVE)** 的视频源是主播实时推送的。因此，主播停止推送后，播放端的画面也会随即停止，而且由于是实时直播，所以播放器在播直播 URL 的时候是没有进度条的。
- **点播 (VOD)** 的视频源是云端的一个视频文件，只要未被从云端移除，视频就可以随时播放，播放中您可以通过进度条控制播放位置，腾讯视频和优酷、土豆等视频网站上的视频观看就是典型的点播场景。

协议的支持

通常使用的直播协议如下，标准直播推荐使用 FLV 协议的直播地址（以 `http` 开头，以 `.flv` 结尾），快直播使用 WebRTC 协议，更多信息请参见：[快直播拉流](#)

直播协议	优点	缺点	播放延迟
FLV	成熟度高、高并发无压力	需集成 SDK 才能播放	2s - 3s
RTMP	延迟较低	高并发情况下表现不佳	1s - 3s
WebRTC	延迟最低	需集成 SDK 才能播放	< 1s

说明：

标准直播与快直播计费价格不同，更多计费详情请参见 [标准直播计费](#) 和 [快直播计费](#)。

特别说明

视频云 SDK 不会对播放地址的来源做限制，即您可以用它来播放腾讯云或非腾讯云的播放地址。但视频云 SDK 中的播放器只支持 FLV、RTMP、HLS (m3u8) 和 WebRTC 四种格式的直播地址，以及 MP4、HLS (m3u8) 和 FLV 三种格式的点播地址。

示例代码

针对开发者的接入反馈的高频问题，腾讯云提供有更加简洁的 API-Example 工程，方便开发者可以快速的了解相关 API 的使用，欢迎使用。

所属平台	GitHub 地址
iOS	Github
Android	Github

对接攻略

步骤1：添加渲染 View

为了能够展示播放器的视频画面，我们第一步要做的就是布局 xml 文件里加入渲染 View：

```
<com.tencent.rtmp.ui.TXCloudVideoView
    android:id="@+id/video_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_centerInParent="true"
    android:visibility="visible"/>
```

步骤2：创建 Player

视频云 SDK 中的 **V2TXLivePlayer** 模块负责实现直播播放功能，并使用 **setRenderView** 接口将它与我们刚刚添加到界面上的 **video_view** 渲染控件进行关联。

```
//mPlayerView 即 step1 中添加的界面 view
TXCloudVideoView mView = (TXCloudVideoView) view.findViewById(R.id.video_view);
//创建 player 对象
V2TXLivePlayer mLivePlayer = new V2TXLivePlayerImpl(mContext);
//关键 player 对象与界面 view
mLivePlayer.setRenderView(mView);
```

步骤3：启动播放

```
String flvUrl = "http://2157.liveplay.myqcloud.com/live/2157_xxxx.flv";
mLivePlayer.startPlay(flvUrl);
```

步骤4：画面调整

- **view：大小和位置**

如需修改画面的大小及位置，直接调整 [step1](#) 中添加的 `video_view` 控件的大小和位置即可。

- **setRenderFillMode：铺满 or 适应**

可选值	含义
V2TXLiveFillModeFill	将图像等比例铺满整个屏幕，多余部分裁剪掉，此模式下画面不会留黑边，但因为部分区域被裁剪而显示不全
V2TXLiveFillModeFit	将图像等比例缩放，适配最长边，缩放后的宽和高都不会超过显示区域，居中显示，画面可能会留有黑边

- **setRenderRotation：视频画面顺时针旋转角度**

可选值	含义
V2TXLiveRotation0	不旋转
V2TXLiveRotation90	顺时针旋转90度
V2TXLiveRotation180	顺时针旋转180度
V2TXLiveRotation270	顺时针旋转270度

```
// 设置填充模式
mLivePlayer.setRenderFillMode(V2TXLiveFillModeFit);
// 设置画面渲染方向
mLivePlayer.setRenderRotation(V2TXLiveRotation0);
```



The long side fits the screen



The short side fits the screen



Landscape

步骤5：暂停播放

对于直播播放而言，并没有真正意义上的暂停，所谓的直播暂停，只是画面冻结和关闭声音，而云端的视频源还在不断地更新着，所以当您调用 `resume` 的时候，会从最新的时间点开始播放，这是和点播对比的最大不同点（点播播放器的暂停和继续与播放本地视频文件时的表现相同）。

```
// 暂停
mLivePlayer.pauseAudio();
mLivePlayer.pauseVideo();
// 继续
mLivePlayer.resumeAudio();
mLivePlayer.resumeVideo();
```

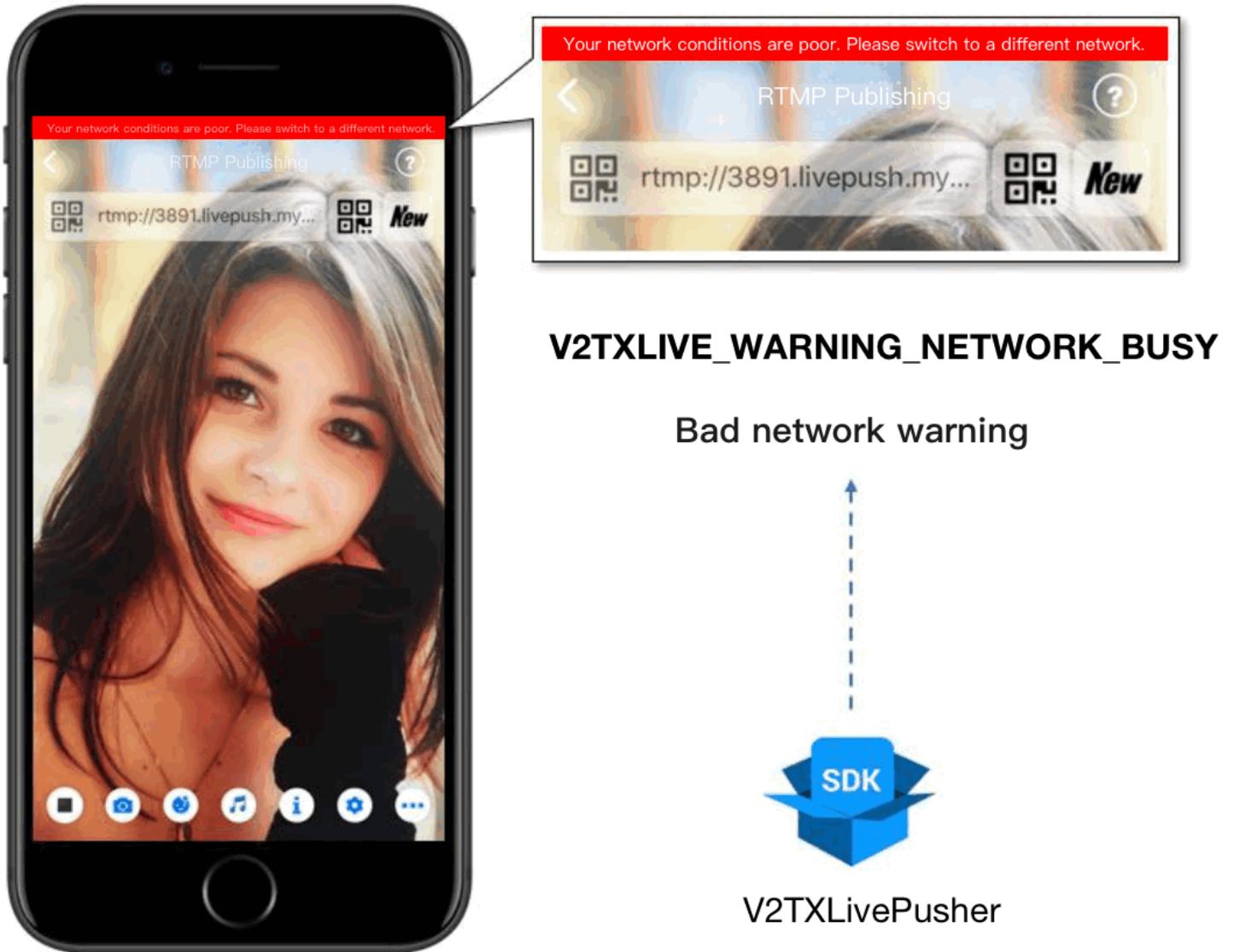
步骤6：结束播放

结束播放非常简单，直接调用 `stopPlay` 即可。

```
mLivePlayer.stopPlay();
```

步骤7：屏幕截图

通过调用 **snapshot** 您可以截取当前直播画面为一帧屏幕，此功能只会截取当前直播流的视频画面，如果您需要截取当前的整个 UI 界面，请调用 Android 的系统 API 来实现。



```

mLivePlayer.setObserver(new MyPlayerObserver());
mLivePlayer.snapshot();
// 在MyPlayerObserver的回调接口onSnapshotComplete中获取屏幕截图
private class MyPlayerObserver extends V2TXLivePlayerObserver {
    ...
    @Override
    public void onSnapshotComplete(V2TXLivePlayer v2TXLivePlayer, Bitmap bitmap) {
    }
    ...
}

```

延时调节

腾讯云 SDK 的云直播播放功能，并非基于 `ffmpeg` 做二次开发，而是采用了自研的播放引擎，所以相比于开源播放器，在直播的延迟控制方面有更好的表现，我们提供了三种延迟调节模式，分别适用于：秀场、游戏以及混合场景。

• 三种模式的特性对比

控制模式	卡顿率	平均延迟	适用场景	原理简述
极速模式	较流畅 偏高	2s- 3s	美女秀场（冲顶大会）	在延迟控制上有优势，适用于对延迟大小比较敏感的场景
流畅模式	卡顿率 最低	>= 5s	游戏直播（企鹅电竞）	对于超大码率的游戏直播（例如绝地求生）非常适合，卡顿率最低
自动模式	网络自 适应	2s-8s	混合场景	观众端的网络越好，延迟就越低；观众端网络越差，延迟就越高

• 三种模式的对接代码

```
//自动模式  
mLivePlayer.setCacheParams(1.0f, 5.0f);  
//极速模式  
mLivePlayer.setCacheParams(1.0f, 1.0f);  
//流畅模式  
mLivePlayer.setCacheParams(5.0f, 5.0f);  
//设置完成之后再启动播放
```

说明：

更多关于卡顿和延迟优化的技术知识，可以阅读 [如何优化视频卡顿？](#)

SDK 事件监听

您可以为 `V2TXLivePlayer` 对象绑定一个 `V2TXLivePlayerObserver`，之后 SDK 的内部状态信息例如播放器状态、播放音量回调、音视频首帧回调、统计数据、警告和错误信息等会通过对应的回调通知给您。

定时触发的状态通知

- **onStatisticsUpdate** 通知每2秒都会被触发一次，目的是实时反馈当前的播放器状态，它就像汽车的仪表盘，可以告知您目前 SDK 内部的一些具体情况，以便您能对当前网络状况和视频信息等有所了解。

评估参数	含义说明
appCpu	当前 App 的 CPU 使用率 (%)
systemCpu	当前系统的 CPU 使用率 (%)
width	视频宽度
height	视频高度
fps	帧率 (fps)
audioBitrate	音频码率 (Kbps)
videoBitrate	视频码率 (Kbps)

- **onPlayoutVolumeUpdate** 播放器音量大小回调。这个回调仅当您调用 `enableVolumeEvaluation` 开启播放音量大小提示之后才会工作。回调的时间间隔也会与您在设置 `enableVolumeEvaluation` 的参数 `intervalMs` 保持一致。

非定时触发的状态通知

其余的回调仅在事件发生时才会抛出来。

快直播拉流

最近更新时间：2022-06-14 13:01:57

快直播概述

快直播（Live Event Broadcasting, LEB）是标准直播在超低延迟播放场景下的延伸，比传统直播协议延迟更低，为观众提供毫秒级的极致直播观看体验。能够满足一些对延迟性能要求更高的特定场景需求，例如在线教育、体育赛事直播、在线答题等。

说明：

- 上图为快直播和标准的 CDN 直播的真实对比视频（使用 [scrcpy 工具](#) 配合录制），从左至右分别为：标准的 CDN 直播、**快直播**、推流端。
- 标准直播与快直播计费价格不同，更多计费详情请参见 [标准直播计费](#) 和 [快直播计费](#)。

方案优势

优势	说明
毫秒级超低延迟播放	采用 UDP 协议将传统直播中3秒 - 5秒延迟降低至1秒以内，同时兼顾秒开、卡顿率等核心指标，给用户带来极致的超低延迟直播体验。
功能完善，平滑兼容	兼容了标准直播包括推流、转码、录制、截图、鉴黄、播放等全功能，支持客户从现有的标准直播业务平滑迁移。
简单易用，安全可靠	采用标准协议，对接简单，在 Chrome 和 Safari 浏览器中无需任何插件即可进行播放。播放协议默认加密，更加安全可靠。

适用场景

场景	说明
体育赛事	快直播为体育赛事提供超低延迟的直播能力加持，使比赛赛事结果快速通过直播触达用户，让观众享受实时了解赛事动态的乐趣。
电商直播	电商直播中，商品拍卖、促销抢购等交易反馈对直播实时性要求很高，快直播的超低延迟能力，能让主播和观众能够及时得到交易反馈，提升边看边买的体验。
在线课堂	师生通过直播完成在线的课堂教学，得益于快直播的超低延迟能力，使课堂互动能力得到提升，让在线课堂也能像线下授课一样自然。

场景	说明
在线答题	传统的在线答题由于存在延迟，观众端有时需要进行补帧才能让观众主持两端同时显示。快直播的超低延迟能够完美解决这个问题，让双方实时看到答题画面，降低了实现难度，也让体验更加流畅。
秀场互动	快直播适用于秀场直播场景，极大优化了在观众送礼等对画面实时性要求高的直播互动场景中的观众互动体验。

体验快直播

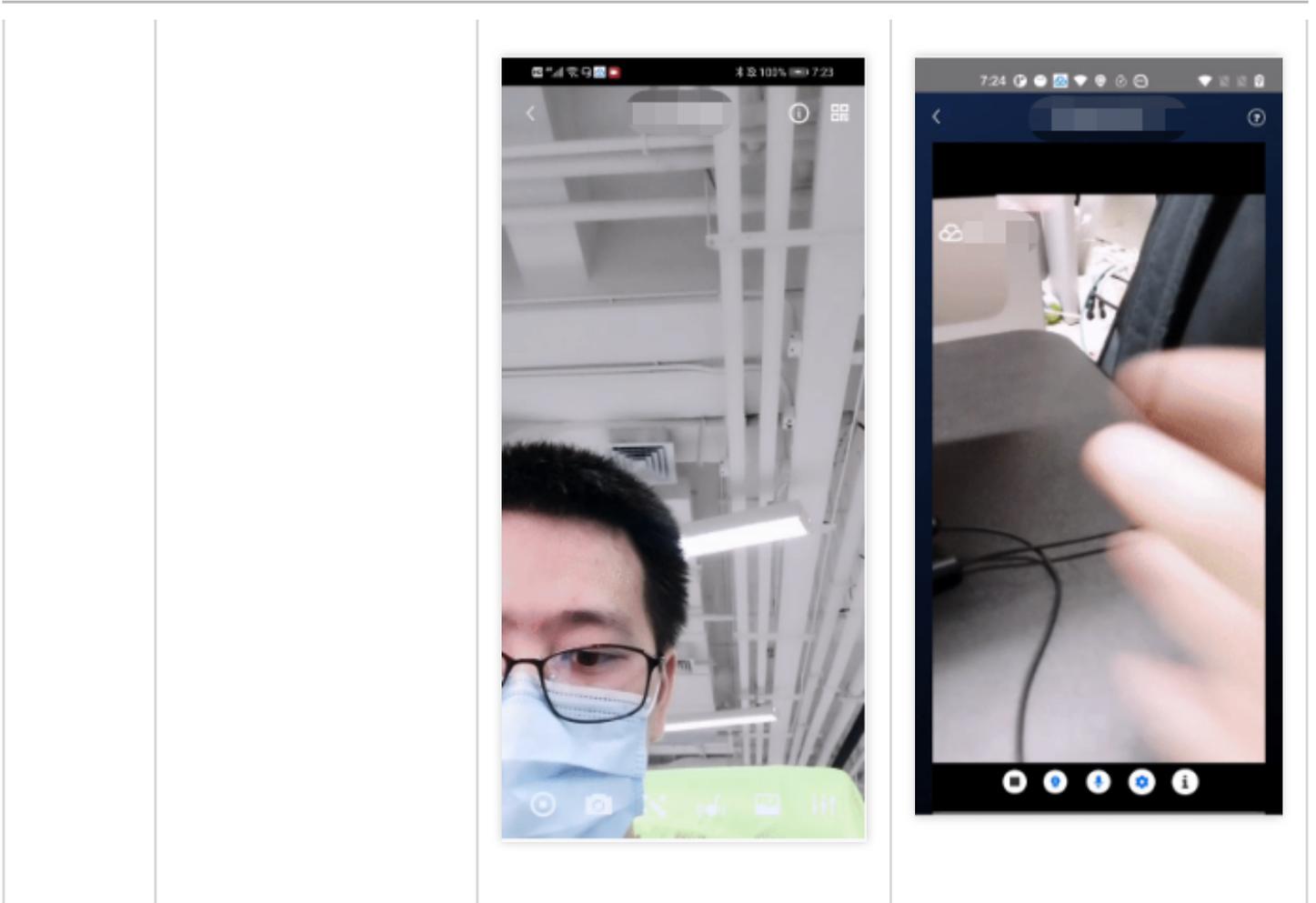
视频云工具包是腾讯云开源的一套完整的音视频服务解决方案，包含实时音视频（TRTC）、移动直播（MLVB）、短视频（UGC）等多个 SDK 的能力展示，其中包含快直播相关体验 UI — **快直播播放**。

说明：

Demo 演示和体验步骤以 Android 为例，iOS Demo 界面略有不同。

源码及示例

源码下载	体验安装	推流演示 (Android)	播放演示 (Android)
Android			
iOS			



说明：

除上述示例外，针对开发者的接入反馈的高频问题，腾讯云提供更加简洁的 API-Example 工程，方便开发者可以快速的了解相关 API 的使用，欢迎使用。

- iOS：[MLVB-API-Example](#)
- Android：[MLVB-API-Example](#)

推流体验

快直播兼容了标准直播，因此可以使用普通推流端进行推流，然后使用快直播进行拉流。

1. 下载 [视频云工具包](#)，安装登录后，进入【推流演示（摄像头推流）】中。
2. 允许相关权限申请，单击【自动生成推流地址】，即开始推流了。
3. 成功推流之后，点击右上角的二维码图标，可以获取播放地址，其中【快直播】就是用于快直播的播放地址。
4. 成功开始推流后，可单击右下侧的菜单按钮，进行美颜、BGM、切换摄像头等设置操作。

播放体验

1. 下载 [视频云工具包](#)，安装登录后，进入【快直播播放】中。

2. 允许相关权限申请，单击二维码扫描按钮，扫描【推流体验】中得到的快直播播放地址。
3. 扫描完成后即开始播放，播放成功后，可单击右下侧的菜单按钮，进行静音、设置等操作。

接入工程

新版本的移动直播 SDK，可以使用 `V2TXLivePlayer` 来播放快直播的流，同时也提供了 `V2TXLivePusher` 来推流。快直播直播协议支持 WebRTC 标准协议，使用标准的扩展方式，其 URL 均以 `webrtc://` 字符开始。

步骤1：下载 SDK

可以在 [SDK 下载](#) 页面选择全功能版或者企业版下载。

步骤2：获取播放 URL

在直播场景中，不论是推流还是拉流都离不开对应的 URL。请参见 [快直播快速入门](#) 获取快直播的播放 URL。快直播 URL 均以 `webrtc://` 字符开始，类似于这样：

```
webrtc://{Domain}/{AppName}/{StreamName}
```

在上述的 URL 中，存在一些关键字段，关于其中关键字段的含义信息，详见下表：

字段名称	字段含义
webrtc://	快直播 URL 的前缀字段
Domain	快直播播放域名
AppName	应用名称，指的是直播流媒体文件存放路径，默认云直播会分配一个路径：live
StreamName	流名称，指每路直播流唯一的标识符

说明：

如果需要推流，具体操作请参见 [摄像头推流](#) 或者 [录屏推流](#)。

步骤3：实现快直播播放

使用 `V2TXLivePlayer` 对象可以使用快直播进行拉流，具体做法如下（传入正确的 URL 是关键）：

示例代码

- [Android](#)
- [iOS](#)

```
// 创建一个 V2TXLivePlayer 对象；
V2TXLivePlayer player = new V2TXLivePlayerImpl(mContext);
player.setObserver(new MyPlayerObserver(playerView));
player.setRenderView(mSurfaceView);
// 传入低延时协议播放地址，即可开始播放；
player.startPlay("webrtc://{Domain}/{AppName}/{StreamName}");
```