

## 推流功能

---

解压 SDK 开发包，进入 Push 目录，参考 PublishViewController 对推流功能的使用：

1. 创建一个 TXLivePush 对象（目前该对象只能被单例使用）
2. 使用 startPreivew 接口函数用来预览摄像头画面。
3. 使用 setVideoQuality 接口函数设置画质级别，普通直播可以设置为 VIDEO\_QUALITY\_HIGH\_DEFINITION
4. 使用 startPush 接口函数用来推流，推流 URL 可以在开通腾讯云 [直播](#) 服务后获取。
5. 美颜滤镜、水印图标、切后台的垫片推流、背景音乐、耳返混响 等等一系列高级功能，推荐大家关注我们的 [官方文档](#)。

## 直播点播

---

直播中视频源是正在推流的主播，点播的视频源则是服务器上已存在的视频文件（e.g.优酷土豆），注意两者的区别。

解压 SDK 开发包，进入 Play 目录，参考 PlayViewController 对播放功能的使用，PlayViewController 同时支持直播播放（界面上没有进度条）和点播播放（界面有进度条）两种能力：

1. 创建 TXLivePlayer 对象（该对象支持多例，但每个 Player 都要有自己独立的 View）。
2. 使用 setupVideoWidget 接口函数指定渲染视频画面用的 UIView。
3. 使用 startPlay 开启播放即可。具体采用什么播放协议，由 startPlay 来支持，LIVE\_ 打头的是直播协议，VOD\_ 打头的是点播地址，LOCAL\_VIDEO 是用来播放本地视频的。
4. 截图、截视频（把视频一部分截取下来压成 MP4）等高级特性，推荐关注我们的 [官方文档](#)。

## 主播连麦

---

连麦指的是直播观众（或者其它房间的主播）可以跟当前主播进行 **实时音视频通话**，同时服务端可以进行 **多路混流**，让观众看到多路画面。

- **示例代码**

解压 SDK 开发包，进入 LinkMic 目录，该目录下的代码演示了如何用 SDK 构建双向或者多人的实时通话场景。

- LinkMicViewController - 实现的是大画面的主体逻辑。
- LinkMicPlayItem - 实现的是小画面的逻辑。
- StreamUrlScanner - 用于加入新的小画面的播放地址。

- **实时通话**

- 实时通话场景中，每一个人都是一路上行（将自己的声音和画面推到云端）和多路下行（从云端拉取其他人的声音和画面）。
- 上行使用 TXLivePush 模块实现，但需要开启回音消除（TXLivePushConfig::enableAEC），下行亦使用 TXLivePlayer 模块实现，同样需要开启回音消除。
- 上下行协议均使用 rtmp 协议，但跟普通 CDN 拉取的音视频流有所不同，连麦中的大小主播使用的均是核心 IDC 机房的超级链路，从服务器到手机的延迟大约平均只有几十毫秒。

- 超级链路的 URL 拼装方式不同于普通的 CDN 播放地址，需要参考 [官方文档](#) 进行拼装。
- 多路混流
- 可以使用腾讯云端 [API](#) 指定将几条视频流混合起来，API 可以指定混流的各种参数，比如小画面的位置等等。

## 视频录制

---

解压 SDK 开发包，进入 VideoRecord 目录了解视频录制功能的用法：

1. 创建 TXUGCRecord 对象，TXUGCSimpleConfig 可用于指定短视频画质、水印等配置项。
2. 使用 startRecord 接口函数启动录制，stopRecord 接口函数结束录制。
3. 录制下来的 MP4 文件和视频封面，会通过 TXVideoRecordListener 回调通知出来。
4. 使用 TXUGCPublish 可以将视频发布到指定的云平台。

更多详细信息，推荐关注我们的 [官方文档](#)。

## 视频编辑

---

视频编辑器具有比较复杂的交互逻辑，这也决定了其 UI 复杂度很高，所以我们比较推荐复用 SDK 开发包中的 UI 源码，VideoEditor 目录包含短视频编辑器的 UI 代码。

如果要自己定制 UI 界面，推荐关注我们的 [官方文档](#)。

## 视频拼接

---

视频拼接器具有比较复杂的交互逻辑，这也决定了其 UI 复杂度很高，所以我们比较推荐复用 SDK 开发包中的 UI 源码，VideoJoiner 目录包含短视频拼接器的 UI 代码。

如果要自己定制 UI 界面，推荐关注我们的 [官方文档](#)。