

短视频 技术支持 产品文档



腾讯云

【版权声明】

©2013-2018 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

技术支持

如何联系我们

错误码

编译问题

Android

上传问题

功能答疑

其他常见问题

Android

iOS

技术支持

如何联系我们

最近更新时间：2018-09-19 11:34:12

热线电话

当您在使用腾讯云遇到问题时，可以直接致电客服人员，寻求相应的帮助。

- 客服咨询：4009-100-100
- 备案帮助：4009-100-100 转 3

工单系统

当您遇到运维或技术类产品使用问题时，可以登录腾讯云官网，根据界面指引提交工单，我们将尽快响应，期待收到您的宝贵意见。

工单相关入口如下：

- 工单提交：[提交工单](#)
- 状态查询：[工单列表](#)

工单状态说明如下：

- 未处理：新提交的工单，或者技术支持收到工单但评估还未完成。您可以对未处理工单进行补充和关闭操作。
- 处理中：技术支持收到工单并评估，正在实施过程中。您无法对处理中的工单做任何操作。
- 待补充：技术支持收到工单并评估，未提交完整的信息，需要补充。您可以对待补充工单进行关闭操作。

说明：当“待补充”的工单补充后重新提交，会再次进入“未处理”状态。

- 已结单：工单实施完成，或者您在技术支持操作前关闭工单。

问答社区

问答社区是腾讯云为开发者提供的交流平台。该平台汇聚了云计算、人工智能、小程序、产品文档等热门标签版块。

您可以登录问答社区进行提问，会有产品专家或热心用户对您提出的问题，进行相应解答、探讨和指导。

平台入口：[问答](#)

技术交流

关注公众号“腾讯云视频”，在聊天窗口发送关键字“技术支持”，将有专人和您联系。



错误码

最近更新时间：2018-06-04 10:18:49

错误码表

短视频录制

code	TXRecordCommon中对应常量	含义说明
-1	START_RECORD_ERR_IS_IN_RECORDING	开始录制时存在未完成的任务
-2	START_RECORD_ERR_VIDEO_PATH_IS_EMPTY	开始录制时视频文件路径为空
-3	START_RECORD_ERR_API_IS_LOWER_THAN_18	版本小于18
-4	START_RECORD_ERR_NOT_INIT	开始录制时还未初始化结束
-5	START_RECORD_ERR_LICENCE_VERIFICATION_FAILED	licence校验失败
-1	RECORD_RESULT_FAILED	录制失败

短视频编辑

code	TXVideoEditConstants中对应常量	含义说明
-100001	ERR_SOURCE_NO_FOUND	通过setVideoPath传入的源视频文件找不到
-100002	ERR_SOURCE_DAMAGED	通过setVideoPath传入的源视频文件已被损坏
-100003	ERR_SOURCE_NO_TRACK	解封装（把合成的视频文件中的视频、音频提取成不同的格式文件）目前不支持的视频格式(OppoR9s目前不支持音频为mp3的音频格式)

短视频上传

SDK错误码

短视频上传SDK 通过 TXRecordCommon.ITXVideoPublishListener 接口来监听短视频发布相关的状态。因此，可以利用TXRecordCommon.TXPublishResult 中的 retCode 来确认视频发布的情况。

code	事件定义	含义说明
0	NO_ERROR	发布成功
1001	ERR_UGC_REQUEST_FAILED	请求上传失败

code	事件定义	含义说明
1002	ERR_UGC_PARSE_FAILED	请求信息解析失败
1003	ERR_UPLOAD_VIDEO_FAILED	上传视频失败
1004	ERR_UPLOAD_COVER_FAILED	上传封面失败
1005	ERR_UGC_FINISH_REQUEST_FAILED	结束上传请求失败
1006	ERR_UGC_FINISH_RESPONSE_FAILED	结束上传响应错误
1007	ERR_USER_CANCEL	用户取消发布操作
1007	ERR_CLIENT_BUSY	客户端正忙(对象无法处理更多请求)
1008	ERR_FILE_NOEXIT	上传文件不存在
1009	ERR_UGC_PUBLISHING	视频正在上传中
1010	ERR_UGC_INVALID_PARAM	上传参数设置有误
1011	ERR_UGC_INVALID_SECRETID	视频上传secretID错误, 已经废弃, 不会抛出
1012	ERR_UGC_INVALID_SIGNATURE	视频上传signature错误
1013	ERR_UGC_INVALID_VIDOPATH	视频文件的路径错误
1014	ERR_UGC_INVALID_VIDEO_FILE	当前路径下视频文件不存在
1015	ERR_UGC_FILE_NAME	视频上传文件名太长或含有特殊字符
1016	ERR_UGC_INVALID_COVER_PATH	视频文件封面路径不对, 文件不存在

服务器错误码

如果您无法从 SDK 错误码中诊断出视频发布的结果, 可以查询以下服务器返回的错误码, 这些错误码可以在 Log 信息中找到。

错误码	含义
0	发布成功
-20001	秒传成功
-20002	任务取消
-20003	任务暂停

错误码	含义
-20004	文件不存在
-20007	服务器回包为空
-20008	请求超时
-20009	appid为空
-20010	bucket为空
-20011	COS远程路径为空
-20012	COS目录中含有保留字符
-20013	dest_fileId为空
-20014	bucket_authority为空
-21001	Out Of Memory
-22000	IO异常
-25000	其他错误，如需处理请联系腾讯商务人员或者提交工单，联系电话：4009-100-100

编译问题

Android

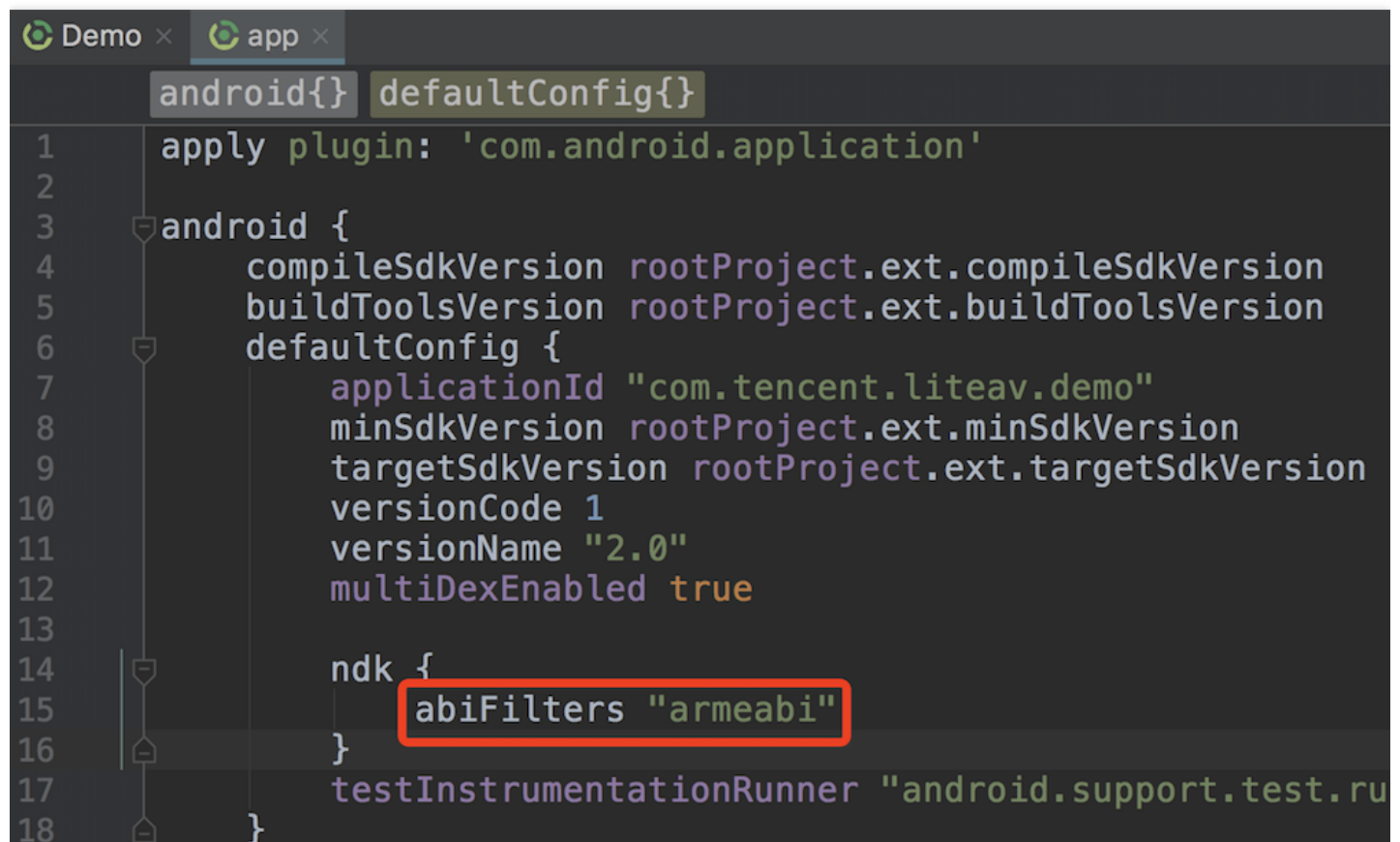
最近更新时间：2018-06-04 10:10:41

编译问题

集成遇到异常

```
java.lang.UnsatisfiedLinkError: No implementation found for void com.tencent.liteav.basic.log.TXCLog.nativeLogInit() (tried
    at com.tencent.liteav.basic.log.TXCLog.nativeLogInit(Native Method)
    at com.tencent.liteav.basic.log.TXCLog.init(TXCLog.java:31)
    at com.tencent.liteav.basic.log.TXCLog.setConsoleEnabled(TXCLog.java:52)
    at com.tencent.rtmp.TXLiveBase.setConsoleEnabled(TXLiveBase.java:73)
```

如果您使用的是 商用企业版，那么您只能使用 armeabi 架构，关闭其他架构，例如armeabi-v7a；如果您使用的是其他版本，那么您可以使用 armeabi 和 armeabi-v7a 架构。



```
Demo x app x
android{} defaultConfig{}
1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion rootProject.ext.compileSdkVersion
5     buildToolsVersion rootProject.ext.buildToolsVersion
6     defaultConfig {
7         applicationId "com.tencent.liteav.demo"
8         minSdkVersion rootProject.ext.minSdkVersion
9         targetSdkVersion rootProject.ext.targetSdkVersion
10        versionCode 1
11        versionName "2.0"
12        multiDexEnabled true
13
14        ndk {
15            abiFilters "armeabi"
16        }
17        testInstrumentationRunner "android.support.test.ru
18    }
```

如上图所示，请在app的build.gradle中指定abiFilters为“armeabi”

SDK升级后，短视频的功能不能使用？

1、如果使用的是androidstudio，在替换新的aar后，请修改app的build.gradle中的aar引用，是否与您放入工程下/libs目录下的aar**文件名称是否一致**。然后重新clean并且build一下您的工程。

2、确认sdk版本，短视频sdk4.5版本之后需要licence支持

请先申请licence，SDK 有两种版本和两种授权 LICENSE，

- SDK 版本分为基础版和商业版，区别在于AI特效的有无；
- LICENSE 分为基础版和商用版，基础版需要申请基础功能的license；商业版除了要申请基础功能的license外，还需要申请AI动效的license；两种都可申请试用版的license。
- 详细价格请参考[文档](#)

上传问题

最近更新时间：2018-06-04 10:11:12

上传问题

客户端视频上传，是指 App 的最终用户将本地视频直接上传到腾讯云点播。

视频上传功能TXUGCPublish找不到？

视频上传模块从 SDK 中独立出来，并开源到 Demo 中，需要客户去集成短视频上传。

1. 下载[Demo](#)
2. 将短视频上传jar(在Demo\app\libs目录下)拷贝到 您的项目\app\src\main\libs 目录下
3. 将短视频上传源代码目录(在 Demo\app\src\main\java\com\tencent\liteav\demo\videoupload 目录下)拷贝到您自己的工程目录下，并修改源代码里面的package名称。
4. 在工程 App 目录下的 build.gradle 中，添加引用 jar 包的代码。

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
}
```

1. 在 AndroidManifest.xml 中配置 App 的权限

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />  
<uses-permission android:name="android.permission.READ_PHONE_STATE" />  
<uses-permission android:name="android.permission.CALL_PHONE"/>  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.READ_LOGS" />  
<uses-permission android:name="android.permission.RECORD_AUDIO" />  
<uses-permission android:name="android.permission.CAMERA" />  
<uses-feature android:name="android.hardware.Camera"/>  
<uses-feature android:name="android.hardware.camera.autofocus" />
```

上传失败1000内部错误？

请检查是否开启点播服务

短视频上传参数类错误

请检查视频文件地址和图片地址是否正确，该路径下是否能找到相应文件

短视频上传签名错误

客户端在发起上传前，需要向 App 服务器请求上传签名,如果 App 服务器允许客户端上传，则应按照[签名规则](#)为客户端生成一个上传签名。客户端执行上传操作时，必须携带该签名，让腾讯云点播验证客户端的上传是否被授权。

客户端上传签名的生成包括以下三步，详情参见[文档](#)：

1. 获取 API 密钥；
2. 拼接明文串；
3. 将明文串转为最终签名。
4. 服务搭建完毕之后，开发者可以通过腾讯云点播提供的工具来校验签名的正确性：
5. [签名生成工具](#)：根据参数和密钥，快速生成签名；
6. [签名校验工具](#)：对签名进行解析，得到生成签名时所使用的各项参数。

功能答疑

最近更新时间：2018-10-08 16:53:35

功能答疑

根据客户的反馈，我们精心准备了一些常见问题答疑，后续会不断更新，希望能帮你解决产品功能的疑问。

1. SDK支持的系统版本？

iOS：手机8.0 以上系统，Xcode 9 或更高版本，OS X 10.10 或更高版本

Android：手机4.0.3 (API 15) 及以上系统，注意只有 (Android 4.3) API 18 以上的系统才能开启硬件编码

2. SDK是否支持X86？

iOS：支持

Android:不支持

3. SDK录制是否支持防抖？

不支持

4. 老版本的SDK是否存在（比如3.9之类的）？

不存在，建议你使用最新版本。

5. 接入短视频是否需要视听证？

不需要。

6. Demo和小视频里面的BGM客户是否可以使用？

不可以，Demo和小视频里面的BGM仅用于功能展示用，如果您用于商业APP，会存在法律风险。

7. 视频录制和编辑是否支持转GIF？

不支持，您可以根据我们SDK接口 (TXVideoEditor -> TXVideoInfoReader -> getSampleImages) 获取视频采样图列表，然后自行生成GIF。

8. SDK是否支持拍照功能？

支持，您可以在录制API 里面调用拍照接口拍照 (TXUGCRecord -> snapshot)。

9. SDK生成的视频是否可以直接上传到非腾讯云平台（比如微信公众号）？

SDK生成的视频可以上传到腾讯云的点播服务器，Demo里面也有对应的源码参考，如果您要上传到其他平台，请自行查看其他平台的上传要求。

10. SDK是否支持小程序？

不支持

11. SDK是否支持大眼瘦脸？

仅商业版本支持

12. SDK怎么获取视频信息（如宽高之类的）？

TXVideoEditor -> TXVideoInfoReader ->getVideoInfo

13. 视频编辑是否支持在视频任意位置插入图片？

不支持

14. sdk解压的密钥是什么？

商业版SDK的解压密钥可以联系跟您对接的商务获取。

精简版和基础版SDK是没有解压密钥的。

其他常见问题

Android

最近更新时间：2018-06-04 10:13:18

其他问题

目前短视频录制生成的分辨率支持自定义吗？有哪些可定制化的输出？

短视频录制可定制参数包括fps(每秒钟有多少帧画面)，GOP（多少秒编出一个关键帧）大小，视频码率(每秒钟编码器产生的音视频数据的多少)，录制最大/最小时长，录制的分辨率以常量方式提供了三种分辨率可供选择：

360x640、540x960、720x1280

解惑：录制为什么以常量方式而不是用户自定义大小，一是以上三种是主流的录制分辨率，二是android手机兼容问题，不支持一些非主流的分辨率，会产生一些花屏、绿屏、马赛克。

通过调用TXUGCRecord类的startCameraCustomPreview接口，将自定义录制的参数传入，代码如下所示：

```
// 自定义配置
TXRecordCommon.TXUGCCustomConfig customConfig = new TXRecordCommon.TXUGCCustomConfig();
customConfig.videoResolution = TXRecordCommon.VIDEO_RESOLUTION_540_960;
customConfig.minDuration = mMinDuration; // 最小时长
customConfig.maxDuration = mMaxDuration; // 最大时长
customConfig.videoBitrate = mBiteRate; // 视频码率
customConfig.videoGop = mGop; // GOP大小
customConfig.videoFps = mFps; // FPS
customConfig.isFront = mFront; // 是否前置摄像头
mTXCameraRecord.startCameraCustomPreview(customConfig, mVideoView);
```

Android短视频录制结束，为什么没有收到onRecordComplete回调？

开始录制短视频前，请先通过调用TXUGCRecord类的setVideoRecordListener()接口设置录制回调的监听器结束时，需要调用TXUGCRecord类的stopRecord()接口结束录制。

```
// 录制前
mTXCameraRecord = TXUGCRecord.getInstance(this.getApplicationContext());
mTXCameraRecord.setVideoRecordListener(this);

...

// 结束录制
mTXCameraRecord.stopRecord();
```

短视频退出录制，开启第二次录制，如何继续接着上一次内容录制？

Demo在onRecordComplete回调之后，调用了mTXCameraRecord.getPartsManager().deleteAllParts()，用于清除分片文件，因为stopRecord已经将分片文件合成完成。

如果录制退出，继续上次录制，不需要删除分片，不要调用mTXCameraRecord.getPartsManager().deleteAllParts()。

@Override

```
public void onRecordComplete(TXRecordCommon.TXRecordResult result) {
    TXCLog.i(TAG, "onRecordComplete, result retCode = " + result.retCode + ", descMsg = " + result.descMsg + ", videoPath = " + result.videoPath + ", coverPath = " + result.coverPath);
    if (mTXRecordResult.retCode < 0) {
        Toast.makeText(TCVideoRecordActivity.this.getApplicationContext(), "录制失败，原因：" + mTXRecordResult.descMsg, Toast.LENGTH_SHORT).show();
    } else {
        mDuration = mTXCameraRecord.getPartsManager().getDuration(); //录制的总时长
        if (mTXCameraRecord != null) {
            mTXCameraRecord.getPartsManager().deleteAllParts(); //删除多次录制的分片文件
        }
        startPreview(); //进去预览界面
    }
}
```

为什么短视频录制设置背景音没有生效？

设置背景音乐一定要在启动录制（TXUGCRecord的startRecord）接口之前设置才能生效。代码调用顺序参照下面示例：

```
TXRecordCommon.TXUGCSimpleConfig simpleConfig = new TXRecordCommon.TXUGCSimpleConfig();
simpleConfig.videoQuality = TXRecordCommon.VIDEO_QUALITY_MEDIUM;
simpleConfig.minDuration = mMinDuration;
simpleConfig.maxDuration = mMaxDuration;
// 1、首先开启预览
mTXCameraRecord.startCameraSimplePreview(simpleConfig, mVideoView);
// 2、再设置背景音乐的路径并播放背景音乐
mBGMDuration = mTXCameraRecord.setBGM(mBGMPath);
mTXCameraRecord.playBGMFromTime(0, mBGMDuration);
// 3、启动录制(customVideoPath:录制后视频路径, customPartFolder:录制视频的文件夹, customCoverPath:录制后视频的封面路径)
int result = mTXCameraRecord.startRecord(customVideoPath, customPartFolder, customCoverPath);
```

录制是否有拍照功能？

短视频SDK有拍照功能，调用TXUGCRecord类的snapshot接口，以TXRecordCommon.ITXSnapshotListener回调异步返回拍照的图片，代码示例如下：

```
private void snapshot() {
    if (mTXCameraRecord != null) {
        mTXCameraRecord.snapshot(new TXRecordCommon.ITXSnapshotListener() {
            @Override
            public void onSnapshot(Bitmap bmp) {
                // 拍照的图片
                saveBitmap(bmp);
            }
        });
    }
}
```

变速录制速度的倍数是多少？

变速录制不支持自定义速度

定义	TXRecordCommon中对应常量	倍数
极慢速	RECORD_SPEED_SLOWEST	0.5倍
慢速	RECORD_SPEED_SLOW	0.8倍
标准	RECORD_SPEED_NORMAL	1倍
快速	RECORD_SPEED_FAST	1.25倍
极快速	RECORD_SPEED_FASTEST	1.5倍

变速录制通过调用TXUGCRecord的setRecordSpeed(record)，设置不同的录制速度

```
mTXCameraRecord.setRecordSpeed(TXRecordCommon.RECORD_SPEED_FAST);
```

导入视频的格式要求？是否支持导入分辨率大于720P（例如2K，4K）的视频？导入文件限制有具体大小吗？

导入视频目前android端仅支持mp4，分辨率没有做限制，导入文件不限制大小。

解惑：

- 导入视频的分辨率没有做限制，不管原视频多大，经过预处理后最大是720P。
- 为了快速的导入视频，SDK4.7后可以不经过预处理（一些功能收到限制，倒放，单针预览等），对于分辨率大于720P的视频，建议需要加上预处理，因为预览是将每一帧解码，一些手机的性能不好，导致解码一帧并渲染到界面的时间过长，导致卡顿。

目前短视频编辑支持哪种格式的背景音乐？

目前仅支持 mp3 和 M4A 类型。

目前短视频编辑有哪些可定制化的输出？

短视频编辑可定制视频码率（SDK4.5及以上）、音频码率（SDK4.7及以上）、分辨率以常量方式提供了四种分辨率可供选择：360x640、480x640、540x960、720x1280

分辨率	TXVideoEditConstants中对应常量
360x640	VIDEO_COMPRESSED_360P
480x640	VIDEO_COMPRESSED_480P
540x960	VIDEO_COMPRESSED_540P
720x1280	VIDEO_COMPRESSED_720P

```
//设置输出视频码率
mTXVideoEditor.setVideoBitrate(3600);
//设置输出分辨率
mTXVideoEditor.generateVideo(TXVideoEditConstants.VIDEO_COMPRESSED_720P, mVideoOutputPath);
```

通过短视频录制功能录制的视频，其中的音频是可以剥离出来的吗？

目前短视频录制不支持同时录制BGM和人声，所以进入编辑后，重新设置BGM，可以将原声音量设为0，达到替换BGM的目的，代码如下所示：

```
// 设置视频原声音量大小（设为0，去掉录制的BGM）
mTXVideoEditor.setVideoVolume(0.0f);
// 设置本地的背景音乐路径
String bgmPath = getBGMPath();
mTXVideoEditor.setBGM(bgmPath);
// 设置背景音乐音量大小，范围0.0f-1.0f
mTXVideoEditor.setBGMVolume(1.0f);
```

预览画面在同一个Activity窗口和全屏模式如何切换？

动态修改传入SDK视频预览View父布局的大小，SDK内部会根据父布局的大小，根据视频宽高动态调整视频的大小。

SDK接口的调用顺序：

先进行stopPlay，在修改传入SDK的FrameLayout的宽高，调用initWithPreview(parm)，将新的FrameLayout承载播放组建的layout传入，再次startPlay。

```
// 停止播放
mTXVideoEditor.stopPlay();
if (isFullScreen) {
// 如果是全屏模式，则下面切换到窗口模式
FrameLayout.LayoutParams params = new FrameLayout.LayoutParams(ViewGroup.LayoutParams.MA
TCH_PARENT, 1500);
mVideoPlayerLayout.setLayoutParams(params);
initPlayerLayout(false);
isFullScreen = false;
} else {
// 如果是窗口模式，则下面切换到全屏模式
FrameLayout.LayoutParams params = new FrameLayout.LayoutParams(ViewGroup.LayoutParams.MA
TCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT);
mVideoPlayerLayout.setLayoutParams(params);
initPlayerLayout(false);
isFullScreen = true;
}
// 开始播放
mTXVideoEditor.startPlayFromTime(startTime, endTime);

// 重新设置预览View
private void initPlayerLayout(boolean isFullScreen) {
TXVideoEditConstants.TXPreviewParam param = new TXVideoEditConstants.TXPreviewParam();
param.videoView = mVideoPlayerLayout;
if (isFullScreen) {
param.renderMode = TXVideoEditConstants.PREVIEW_RENDER_MODE_FILL_SCREEN;
} else {
param.renderMode = TXVideoEditConstants.PREVIEW_RENDER_MODE_FILL_EDGE;
}
mTXVideoEditor.initWithPreview(param);
}
```

短视频编辑时，腾讯云短视频demo是把“剪辑”和“滤镜”等功能放在一个页面处理。不过，我们公司产品是把“剪辑”功能和“滤镜”分成两个页面？

可以先进行裁剪(setCutTimeFrom)+预处理(processVideo)同时执行，结果生成一个裁剪后的视频预处理完的视频，再进行各种编辑的操作，将裁剪设置成整个时长(setCutTimeFrom)，最后调用generateVideo生成视频，防止压缩两次导致画质降低。

注意：在预处理进行裁剪了，生成完的预处理视频，在最后生成前，一定要将裁剪时长设置为整个视频时长，不然还会再次进行裁剪。

```
//裁剪页面
mTXVideoEditor = new TXVideoEditor(mContext);
```

```
mTXVideoEditor.setCutFromTime(mTCVideoEditView.getSegmentFrom(), mTCVideoEditView.getSegmentTo());
mTXVideoEditor.processVideo();

// 将裁剪设置成整个时长(setCutTimeFrom)
mTXVideoEditor.setCutFromTime(0, mVideoDuration);
// 跳转到特效页面, 进行生成
mTXVideoEditor.generateVideo(TXVideoEditConstants.VIDEO_COMPRESSED_720P, mVideoOutputPath);
```

iOS

最近更新时间：2018-07-18 11:16:49

关于 license 的问题

短视频 SDK 分基础版和商业版，从 4.5 版本开始需要 Licence，基础版只需要短视频的 Licence (TXUgcSDK.licence)，商业版还同时需要 Pitu 的 Licence (YTFaceSDK.licence)，把 Licence 放到工程目录，并修改为对应的名字即可。

4.9 版本开始 Licence 使用方式有改变，可以选择是否把 Licence 打包到项目中。使用时需要调用 setLicenceURL: key: 接口设置 Licence 的 url 和 key。

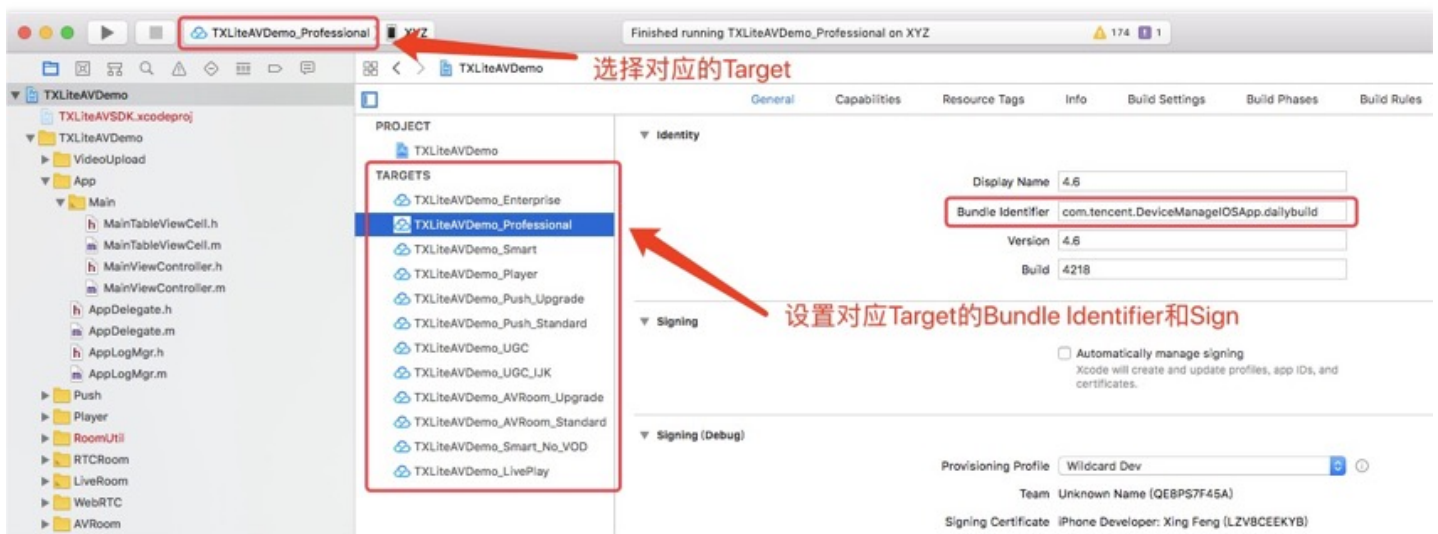
[注意] 4.5 ~ 4.7 版本的 SDK 不支持 Licence 自动续期，4.9 版本开始才支持自动续期。4.9 的 SDK 可以兼容之前的 Licence (url 和 key 不能传 null，可以随便传个字符串)，但是新的 Licence 无法在 4.9 之前的 SDK 上使用。

关于 TXUGCPublish.h 的问题

从 4.5 版本开始，TXUGCPublish 相关的类从 SDK 上移到了 Demo 层，开发者如果需要使用，使用时直接把 VideoUpload 目录整个拖入自己的工程即可。

Xcode 中直接运行 Demo 报错

需要选择对应的 Target，如下图：



连接 Xcode 调试，短视频录制报错

此问题在4.9版本已经修复

连接 Xcode 调试，短视频录制报错，报错信息：Main Thread Checker: UI API called on a background thread

```

15 0x100570d14 <+52>: adrp x8, 3087
16 0x100570d18 <+56>: ldr x1, [x8, #0x1b0]
17 0x100570d1c <+60>: bl 0x100dba91c ; symbol stub for: objc_msgSend
18 0x100570d20 <+64>: mov x29, x29 ; Thread 9: -[UIView layer] must be used from main thread only
19 0x100570d24 <+68>: bl 0x100dba970 ; symbol stub for: objc_retainAutoreleasedReturnValue

Main Thread Checker: UI API called on a background thread: -[UIView layer]
PID: 86092, TID: 8057847, Thread name: (none), Queue name: com.TXC.openGLESContextQueue, QoS: 21
Backtrace:
4 TXLiteAVDemo_Professional 0x0000000100570d20 -[TXCUIElement updateWithView:] + 64
5 TXLiteAVDemo_Professional 0x0000000100547eb0 -[TXCWaterMark setWaterMark:] + 80
6 TXLiteAVDemo_Professional 0x000000010055c1b4 -[TXCVideoPreprocessor setFilterParam] + 3748
7 TXLiteAVDemo_Professional 0x000000010055ac44 __62-[TXCVideoPreprocessor
processFrame:orientation:outputFormat:]_block_invoke + 540
8 libdispatch.dylib 0x0000000102649220 _dispatch_client_callout + 16
9 libdispatch.dylib 0x000000010265be24 _dispatch_sync_invoke_and_complete_recurse + 124
10 libdispatch.dylib 0x000000010265b848 _dispatch_sync_wait + 552
11 TXLiteAVDemo_Professional 0x2018-06-08 17:41:51.498422+0800 TXLiteAVDemo_Professional[86092:8057578] level:1|
    
```

原因是某些 API (一般是UI相关的) 需要在主线程调用，如果在非主线程调用，同时有勾选 Main Thread Checker 的话，就会报错。

解决办法：Product -> Scheme -> Edit Scheme -> Run -> Diagnostics，取消勾选 Main Thread Checker。

使用 SDK 时报找不到头文件

- 在 Build Settings -> Search Paths -> Header Search Paths 中添加头文件搜索路径。
- 使用 "TXLiteAVSDK_UGC/XXX.h" 方式引用 SDK 的头文件。
- 使用 @import TXLiteAVSDK_UGC; 方式引用 SDK (5.0及之后的版本)。

以上几种方法选其一。

运行工程时报找不到类别方法或者 crash

SDK 用到了一些类别的方法，加载类别方法需要在工程配置：Build Settings -> Linking -> Other Linker Flags 添加 -ObjC。

集成 SDK 报 duplicate symbol (符号重复)

- 检查 Build Settings -> Linking -> Other Linker Flags 设置，不能有 -all_load。
- 更换 rename 版 SDK。

录制短视频时设置背景音乐无效

1. 确定传的 BGM path 下有没有文件，以及是否可以正常播放。
2. 确定接口的调用顺序：startCameraSimple:preview: -> setBGM: -> startRecord。

[注意] 很多接口调用有时序要求，不然会无效。一般在注释上会有说明。

比如短视频录制的 `setVideoResolution:`、`setVideoBitrate:`、`setAspectRatio:` 等接口都需要在 `startRecord` 之前设置才有效。

录制设置 BGM 不能循环播放

目前逻辑暂未支持循环播放。

录制设置 BGM，endTime 时没有完成回调

如果设置的 `endTime` 小于音乐文件总时长，4.6 及之前的版本是在 BGM 播放完后才会触发完成回调，从 4.7 版本开始在 `endTime` 时触发完成回调。

为什么录制时第一次打开摄像头比较慢？

苹果手机摄像头第一次打开时（冷启动）耗时相对较长，通过系统接口打开摄像头也是如此。

因为摄像头打开的操作不适合放在子线程去做，经过测试在子线程中进行打开摄像头操作耗时会更大，并且在主线程连续打开/关闭摄像头的时候，子线程的响应延迟会更高，体验不好。

返回继续录制怎么实现？

在第一次录制完成的时候，不要调用 `stopRecord` 和 `stopCameraPreview`（调用之后不能再继续录制，只能重新录制），可以调用 `pauseRecord`，然后通过 `TXUGCPartsManager.getVideoPathList` 获取已经录取的视频片段，通过 `TXVideoJoiner.joinVideo` 合成最终视频（4.5之前版本），还可以直接调用 `TXUGCPartsManager.joinAllParts` 合成最终视频，这个方法合成速度更快（4.5以后版本支持），这样当返回继续录制的时候，所有的录制状态都在，可以继续录制。

短视频录制完成时收不到完成回调

- 确定有没有调用 `stopRecord`，只有调用 `stopRecord` 后才会有完成回调。
- 确定函数的调用是否都在主线程。

录制过程中用其他播放器播放视频，返回继续录制，声音录制不了

iOS 中的 `AudioSession` 是所有音视频应用共用的，使用其他播放器播放的时候，`AudioSession` 会被占用，播放结束时如果 `AudioSession` 没有让出或者让出不及时，会导致录制模块的 `AudioSession` 失效，SDK 提供了 `-(void) pauseAudioSession` 和 `-(void) resumeAudioSession` 两个接口，在去其他播放器预览的时候先调用 `pauseAudioSession`，返回继续录制前调用 `resumeAudioSession`。

为什么录制出来的视频不清晰？

码率和分辨率不匹配，录制出来的视频就会不清晰。可以通过适当增大码率、开启B帧来提升画质。

视频编辑时退后台再回到前台，视频生成失败

生成视频默认采用的是硬编码（编码效率高，编码出来的图像效果好），硬编码器在程序进后台后会停止工作，从而导致视频生成失败。SDK 提供了两个接口 `pauseGenerate` 和 `resumeGenerate`，APP 进后台时可以调用 `pauseGenerate` 暂停视频生成，APP 回到前台后再调用 `resumeGenerate` 继续视频生成。

[注意] 调用 `resumeGenerate`，SDK 将重启硬编码器，有一定的概率重启失败，或重启后前几帧数据编码失败。此时，SDK 内部会在 `TXVideoGenerateListener` 抛出错误事件，收到错误事件后需要重新生成视频。

文件上传失败

文件上传状态码：

```
typedef NS_ENUM(NSInteger, TXPublishResultCode)
{
    PUBLISH_RESULT_OK = 0, //发布成功
    PUBLISH_RESULT_UPLOAD_REQUEST_FAILED = 1001, //step1: "文件上传请求"发送失败
    PUBLISH_RESULT_UPLOAD_RESPONSE_ERROR = 1002, //step1: "文件上传请求"收到错误响应
    PUBLISH_RESULT_UPLOAD_VIDEO_FAILED = 1003, //step2: "视频文件"上传失败
    PUBLISH_RESULT_UPLOAD_COVER_FAILED = 1004, //step2: "封面文件"上传失败
    PUBLISH_RESULT_PUBLISH_REQUEST_FAILED = 1005, //step3: "短视频发布请求"发送失败
    PUBLISH_RESULT_PUBLISH_RESPONSE_ERROR = 1006, //step3: "短视频发布请求"收到错误响应
};
```

1. 确定上传的文件是否在本地沙盒，如果上传媒体库的文件，需要先 copy 到本地沙盒。
2. 返回错误码1002：签名有问题、时间戳过期、点播服务问题（未开通或停服）。
3. 返回错误码1003：请求参数问题、上传文件格式不支持。

短视频录制是否有拍照功能？

短视频 SDK 可以实现拍照功能，开始预览后调用 `TXUGCRecord` 类的 `snapshot` 接口获取图片即可。