

ALIBABA CLOUD

阿里云

视频点播
常见问题

文档版本：20211202

 阿里云

法律声明

阿里云提醒您在使用或阅读本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置>网络>设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

- 1.点播开通常见问题 ----- 05
- 2.媒资上传常见问题 ----- 06
- 3.播放数据统计常见问题 ----- 09
- 4.短视频常见问题 ----- 11
 - 4.1. Android常见问题 ----- 11
 - 4.2. iOS常见问题 ----- 14
 - 4.3. SDK和License常见问题 ----- 16
- 5.播放器问题 ----- 18
 - 5.1. Web播放器常见问题 ----- 18
 - 5.2. 播放器调试和实现流程问题 ----- 22
 - 5.3. 安全文件获取问题 ----- 25
 - 5.4. 参数解析问题 ----- 28
 - 5.5. 短视频SDK在IOS模拟器上安装和使用的方法 ----- 30
- 6.事件通知常见问题 ----- 32
- 7.如何启用H5的同层播放 ----- 34
- 8.HLS标准加密常见问题 ----- 37

1. 点播开通常见问题

本文为您介绍开通视频点播时的问题。

开通点播服务时，会默认帮我开通OSS和CDN产品吗？

不会。视频点播产品在存储、转码、播放加速等服务环节会用到不同的资源。但视频点播作为独立付费产品，无需用户再去开通OSS、MPS、CDN等产品（默认也不会自动为用户开通），为用户提供了私有的视频存储空间，自动发起和完成视频的转码任务，并根据用户提供的加速域名进行播放加速。

视频点播和OSS有什么区别？

从定义上来说：

- 视频点播是集视频音视频采集、编辑、上传、自动化转码处理、媒体资源管理、分发加速于一体的一站式音视频点播解决方案。
- OSS是作为一种海量、安全、低成本、高可靠的云存储服务。

从作用上来说：

- 视频点播中存储的媒体主要是视频，后续也会提供音频等完成媒体库的建设。
- OSS中可以存放各种类型的文件，从视频、音频到图片，再到doc、pdf等等任何文件，可以理解为就是一个云上的硬盘。

从联系上来说：

- OSS下可以作为独立的存储来用，也可以链接CDN产品，完成存放在OSS下文件的分发加速；但视频点播在存储音视频的媒体文件之后，核心作用是提供视频的托管服务，满足在线点播视频的业务需求，并且通过媒资管理、视频在线编辑、视频审核发布、数据统计、短视频SDK、视频安全等能力，覆盖到视频点播业务的不同方面。
- 视频点播的数据存储也是基于OSS Bucket，不同于OSS的是，视频点播为客户提供的私有bucket不占用OSS的空间，无需客户开通OSS产品，该bucket在客户OSS下也不可见，采用独立的计量计费方式。

2.媒资上传常见问题

本文主要介绍媒体上传过程中遇到的页面拉伸，兼容性问题。

微信中使用JS SDK上传存在兼容性问题，无法正常上传

经过排查由于微信浏览器对于H5存在兼容性问题，需要将 `<input type="file" name="file" id="files" multiple="">` 中的 `multiple=""` 参数去掉就可以正常上传。

推流SDK特定分辨率时出现预览页面拉伸现象

推流SDK在选择推流分辨率为480p时预览页面出现拉伸的现象，但是实际推流是正常的。主要因为480p对应的分辨率为480×640，由于大多数手机屏幕均不支持该分辨率的比例导致出现拉伸的现象。

解决办法：修改预览页面surfaceview的比例，请将activity_push.xml内容修改如下即可。

```
public void initView() {  
    mPreviewView = (SurfaceView) findViewById(R.id.preview_view);  
    mPreviewView.getHolder().addCallback(mCallback);  
}
```

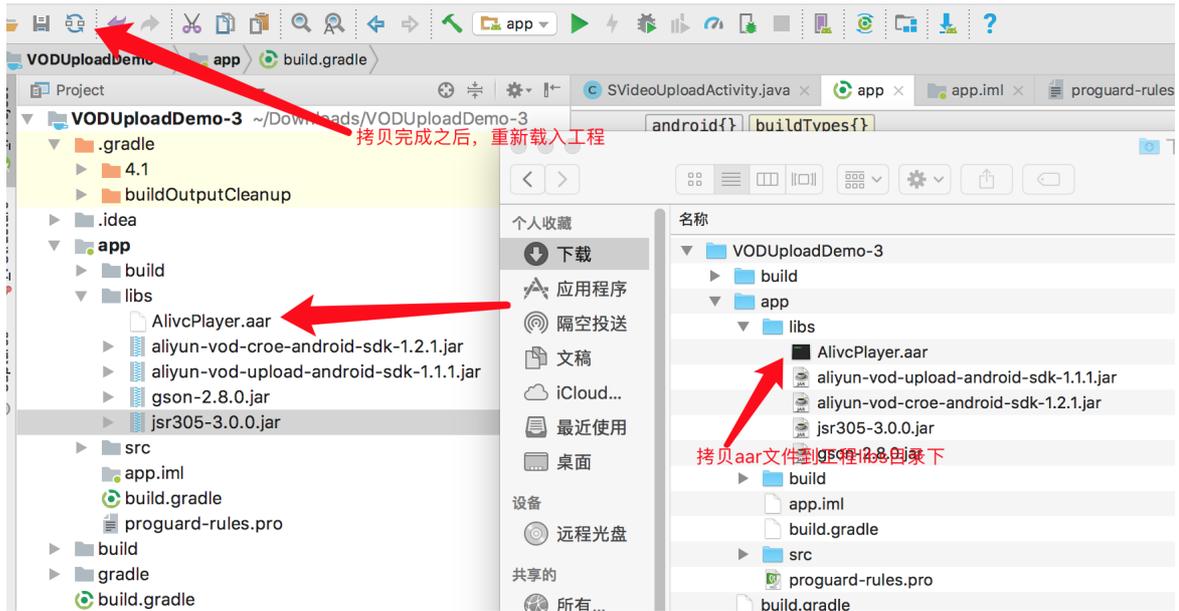
```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <SurfaceView  
        android:id="@+id/preview_view"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"/>  
  
    <!--<FrameLayout-->  
    <!--android:id="@+id/publisher_fragment"-->  
    <!--android:layout_width="match_parent"-->  
    <!--android:layout_height="match_parent"-->  
    <!--android:visibility="gone"/-->  
  
    <android.support.v4.view.ViewPager  
        android:id="@+id/tv_pager"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
    >  
    </android.support.v4.view.ViewPager>  
</RelativeLayout>
```

Android Studio如何查看和导入aar包数据

查看aar包数据：将 .aar 文件后缀改成 .zip 并解压，查看 .class 、 .xml 、 .jar 、 图片、文本等各种内容。

导入aar包数据：

1. 拷贝 .aar 文件到工程项目下，路径一般为 `projectName/libs/`，重新载入工程。



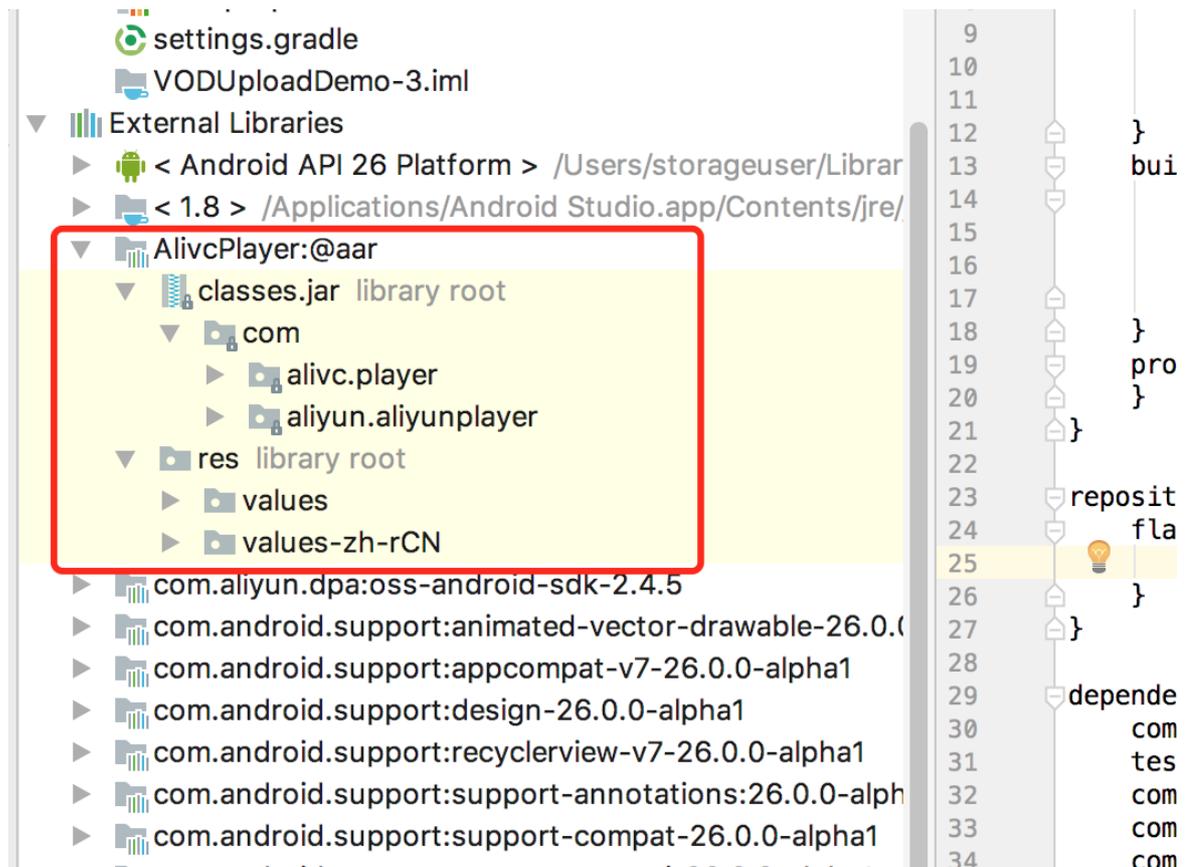
2. 在buid.gradle根标签下添加本地仓库路径，并在dependencies中添加编译依赖。

其中libs目录按照实际工程下的包引入文件夹名称而定。在compile参数中，name的值为aar文件的名称，ext为文件的扩展名。

```
22
23
24 repositories{
25     flatDir{
26         dirs 'libs'
27     }
28 }
29
30 dependencies {
31     compile fileTree(include: ['*.jar'], dir: 'libs')
32     testCompile 'junit:junit:4.12'
33     compile 'com.android.support:appcompat-v7:26+'
34     compile 'com.android.support:design:26+'
35     compile (name:'AlivcPlayer',ext:'aar')
36     //上传SDK 需要依赖OSS上传的SDK
37     compile 'com.aliyun.dpa:oss-android-sdk:2.4.5'
38 }
```

3. 选择 build > rebuild，重新构建project。

构建完成之后，在工程的External Libraries中即可看到引入的aar包。



3.播放数据统计常见问题

本文主要介绍播放数据的统计准确和统计次数等问题。

视频点播的播放数据是如何统计出来的？数据准确吗？

统计方式	准确性	说明
通过播放器SDK的数据埋点来获取播放行为数据	较准确，推荐使用	有完备数据埋点解决方案的播放器（如阿里云所提供的播放器SDK）可以通过提取定期上报的日志信息来统计较为准确、详尽的视频播放行为数据，如视频播放次数、播放时长和观看用户数等。阿里云目前已上线相关服务，支持以官方播放器SDK为数据采集端的视频播放统计，并将一个终端识别为一个用户。基于上述原理，对于未使用阿里云播放器SDK的视频播放行为（包括同时使用SDK和地址播放的用户，其中地址播放的部分），因为难以采集到详尽的播放行为日志来保证数据的准确度，所以目前不提供相关统计。
通过CDN的内容请求次数来估算视频播放次数	不准确，不推荐使用	有的产品可能会通过CDN的内容请求次数来估算视频播放，将每次请求都计为一次播放行为并将这个粗略统计输出给用户。这样的数据是不准确的（比如：请求次数通常大于播放次数，无法区分终端到用户粒度，缺失播放时长、行为信息等），可能会给用户造成潜在损失，因此阿里云目前不提供类似的统计。

我只使用了阿里云的官方播放器SDK作为视频播放渠道，如果我在查询前一天的播放统计，这些数据是“完全”准确的吗？

视频的播放数据统计存在无法避免的合理范围误差，其误差主要来自两个方面：

- 阿里云官方播放器的日志采集间隔是30s，对于用户在视频播放页（而非播放器内）意外中断播放的行为，目前难以合理捕获，因此视频播放的时长统计会存在最大不超过30s的误差值。
- H5播放器会因为用户终端的更新而发生key的变化，从而导致该终端会被识别为一个新用户，进而产生一个新的UV计数，造成播放用户数的统计误差。

我只使用了阿里云的官方播放器SDK作为视频播放渠道，但是从播放数据去估算时，流量和播放次数对不上？

由于采集、处理方式的不同，目前点播视频的播放数据统计和流量统计都有一定的时间延迟，且延迟时间不一致。因此您在查询最新数据时，看到的播放次数和流量数据在同一时间可能无法完全匹配。在查询大于延迟时间窗口的历史数据时（如查询前一天的数据），此影响就会消失。阿里云视频服务团队会尽快改善这一问题，以减轻对您的影响。

除此外，视频的一次播放不等同于视频的一次“完全播放”。用户在观看视频时有可能中断播放、跳跃播放或异常退出，这些行为均会造成合理范围内的数据统计误差。因此，在估算流量时您不能严格按照 $\text{播放次数} \times \text{视频大小}$ 或 $\text{播放次数} \times \text{码率} \times \text{播放时间}$ 来计算。

我使用了阿里云的官方播放器SDK，但也使用了地址播放，为什么我的播放数据统计不准确？

目前阿里云只支持以官方播放器SDK为数据采集端的视频播放统计。对于未使用阿里云官方播放器SDK的视频播放行为（包括同时使用SDK和地址播放的用户，其中地址播放的部分），因为难以采集到详尽的播放行为日志来保证数据的准确度，所以目前不提供相关统计。而任何一次实际的视频播放请求都会产生相应的资源消耗，因此就有播放次数少、流量消耗多的情况发生。

我在没有使用阿里云的官方播放器SDK的情况下，控制台播放次数统计数据始终为0，但是却有流量消耗？

如下面控制台图所示：“只有使用点播播放器才会有播放数据和TOP数据”。对于未使用阿里云官方播放器SDK的播放行为，由于难以保证数据的准确度，目前不提供相关的播放数据统计。因此，您的控制台播放数据为0并不意味着您的视频没有通过其他渠道被播放过，而任何一次实际的视频播放请求都会产生相应的资源消耗。此即您有流量消耗产生却没有播放次数统计的原因。



4.短视频常见问题

4.1. Android常见问题

本文主要介绍短视频SDK在Android端的使用问题。

SDK集成

安卓指令集的兼容情况如何？

目前SDK提供的指令集：armeabi-v7a和arm64-v8。

建议：其他第三方包全部使用armeabi-v7a和arm64-v8的包。如果其他第三方的包没有提供armeabi-v7a或者arm64-v8的包，可以将SDK的so拷贝到armeabi里面。然后使用gradle命令行加入：

```
defaultConfig {
    ...
    ndk {
        abiFilters "armeabi" //如果需要使用armeabi-v7a, 则可将此处armeabi修改为armeabi-v7a
    }
}
```

集成SDK后，Debug版本可以正常运行，但是Release版本启动就崩溃，该怎么解决？

先检查崩溃的log是否是报JNI找不到对应的Java类，如果是的话，一般来说就是混淆导致的，因为JNI调用Java类用的是反射，所以如果混淆把SDK内部与JNI有关的类混淆了，则JNI加载时将无法找到对应的Java类，就会加载失败，解决办法就是把Demo的混淆配置拷贝到开发者工程的混淆配置中。

如何添加硬编黑名单，硬解白名单？

- 添加硬编码黑名单

```
/**
 * 添加硬编码黑名单，model和versions的顺序必须对应起来
 * 黑名单内的机型将使用软编，黑名单外的机型都使用硬编
 * @param models 机型Model信息列表{@link Build#MODEL}
 * @param versions 系统版本号列表{@link Build.VERSION#SDK_INT}，如果不需要适配版本号，填写0即可
 */NativeAdaptiveUtil.encoderAdaptiveList(String[] models,int[] versions);
```

- 添加硬解码白名单

```
/**
 * 添加硬解码器白名单，model和versions的顺序必须对应起来
 * 如果开启硬解码了，则白名单中的机型将使用硬解码，白名单外的机型将使用软解码
 * @param models 机型Model信息列表{@link Build#MODEL}
 * @param versions 系统版本号列表{@link Build.VERSION#SDK_INT}，如果不需要适配版本号，填写0即可
 * @see #setHWDecoderEnable(boolean)
 */
NativeAdaptiveUtil.decoderAdaptiveList(String[] models,int[] versions);
```

Android基础版本提示java.lang.NoSuchFieldError: No field height of type I in class Lcom/aliyun/snap/snap_core/R\$id; or its superclasses (declaration of 'com.aliyun.snap.snap_core.R\$id' appears in /data/app/com.rablive.jwrlive-2/base.apk:classes2.dex), 该如何解决？

原因分析：出现这个错误是因为在开发者的工程中存在和AAR（SDK）一样的xml，于是导致冲突。

解决方法：找到出现冲突的xml。开发者自行加前缀。目前发现容易冲突的xml包括：`activity_setting.xml` 和 `activity_video_play.xml`。

Android基础版提示`java.lang.NoSuchFieldError: No static field notification_template_lines of type I in class Lcom/aliyun/snap/snap_core/R$layout; or its superclasses (declaration of 'com.aliyun.snap.snap_core.R$layout' appears/data/app/com.Aliyun.AliyunVideoSDK.VodSaaSdemo_android-1/base.apk)`，该如何解决？

目前主要原因是基础版SDK的UI不开源，所以内部是引用了support包的，但是打成AAR时是没有将support包打入的，这就导致ID不对应的情况。目前需要您将support版本包对应，如下：

```
//重要：如果工程中引入第三方库也引入了support包，也必须要保证第三方的包版本对应，建议以源码引入第三方库
compile 'com.android.support:appcompat-v7:24.2.1'
compile 'com.android.support:design:24.2.1'
```

但是有时第三方库带support包修改起来比较麻烦，且可能有些第三方包并不是源码引入的，此时建议在Application里面的gradle文件中配置。

```
configurations.all {
    resolutionStrategy {
        force 'com.android.support:appcompat-v7:24.2.1'
        force 'com.android.support:design:24.2.1'
    }
}
```

Demo使用时提示`Please invoke the FileDownloader#init in Application#onCreate first`，该怎么解决？

需要在Application OnCreate中调用`DownloaderManager.getInstance().init(Context context);`。

SDK内部是否有获取视频封面的接口？

Android短视频SDK目前专业版提供了`AliyunIThumbnailFetcher`接口，可以获取非关键帧的图片，其他版本建议使用系统函数取帧。

视频录制

录制如何添加普通动图？

添加普通动图需要使用`AliyunRecorder#addPaster(EffectPaster effectPaster,float sx,float sy,float sw,float sh,float rotation,boolean flip)`接口，并且在`EffectPaster`对象中填入信息，`EffectPaster`的`isTrack`一定要设置为`false`，否则贴图将作为人脸贴图处理，跟随人脸变化，如果没有人脸，贴图不会显示。另外该接口必须在`RecordCallback#OnInitReady()`回调之后调用，否则将不会显示贴图。

如何设置录制角度？

目前设置录制角度有`setRotation`和`setRecordRotation`两个接口，`setRotation`接口是一个自适应接口，只要把手机角度传感器返回的角度值传给这个接口，就可以得到正确的录制角度和人脸角度。`setRecordRotation`接口是自定义的视频角度接口，可以根据您的需求定制任意角度值。

如何实现横屏录制？

- 您如果需要默认横屏只需要将界面UI元素旋转引导横屏拍摄视频，不需要设置界面横屏，让界面固定竖屏即可。

```
android:screenOrientation="portrait"
```

- 横屏拍摄的视频录制完成合成的视频是会带旋转角度的，旋转角度以录制的第一段为准。
- 如果是专业版，在编辑之后调用合成接口合成的视频将会输出一个不带角度的视频。比如原始视频为 360/640，角度：270变为640/*360，由于基础版和标准版只有录制功能，如果横屏拍摄会得到一个带旋转角度的视频，这个视频是以拍摄时第一段的角度为准的。专业版录制时的行为同基础版和标准版，合成完成后视频不带角度，转换为一个角度为0，宽高变换的视频。

关键接口函数：

```
/**
 * 设置视频旋转角度值
 * @param rotation
 */
void setRotation(int rotation);
```

接口调用条件：设置旋转角度需要在初始化完成之后设置，且需要保证录制第一段之前调用。

调用操作步骤：让界面固定竖屏，然后设置旋转角度即可。

1. 设置界面竖屏，让界面的UI元素旋转，引导您拍摄横屏的视频。
2. 同普通录制的初始化。
3. 在调用开始录制前调用，注意旋转角度需要您自己获取，可以参考Demo使用OrientationDetector来获取方向。

```
mRecorder.setRotation(int rotation);
```

4. 继续录制步骤，注意每次调用startRecording前都需要设置旋转角度以此来确定每段视频的旋转角度。

拍摄添加背景音乐，完成后调用finishRecordForEditor，音乐没有合成进去，是什么原因？

添加背景音乐后，必须调用finishRecording接口，才会把音乐合成进去，否则不会合成进去，造成进到编辑界面无音乐的结果。finishRecording和finishRecordForEditor之间的区别如下：

- finishRecording有两个作用，一是录制多段的时候，调用该接口可以将多段拼接成一个mp4，也就是录制指定的输出文件，二是添加了背景音乐后，调用该接口会把背景音乐合成进这个输出mp4中，无论是多段还是单段都可以。
- finishRecordForEditor不会拼接多段视频，也不会将背景音乐合成到输出mp4文件中，而是仅仅将录制的片段（startRecording > stopRecording之后就会生成一个片段），按照指定的格式配置到project.json文件中（创建Editor时传入的Uri就是该文件的Uri）。

如果添加了背景音乐的录制多段，该如何进到编辑界面呢？

先调用finishRecording将视频拼接成输出地址的mp4，然后将该mp4用AliyunIImport接口导入到编辑界面。

录制界面添加普通动图显示效果不完整，该怎么解决？

需要获取普通动图素材的宽和高。

```
AliyunIRecorder#addPaster(EffctPaster effectPaster,float sx,float sy,float sw,float sh,float rotation,boolean flip)
```

其中sw和sh参数需要符合素材宽高和屏幕宽高的比列。

例如：素材的宽高分别为200、300，屏幕的宽高分别为540、720。则 $sw = (float)200/540$, $sh = (float)300/720$ 。sx、sy是归一化的屏幕比例参数，坐标是以资源的中心点作为锚点的。

视频编辑

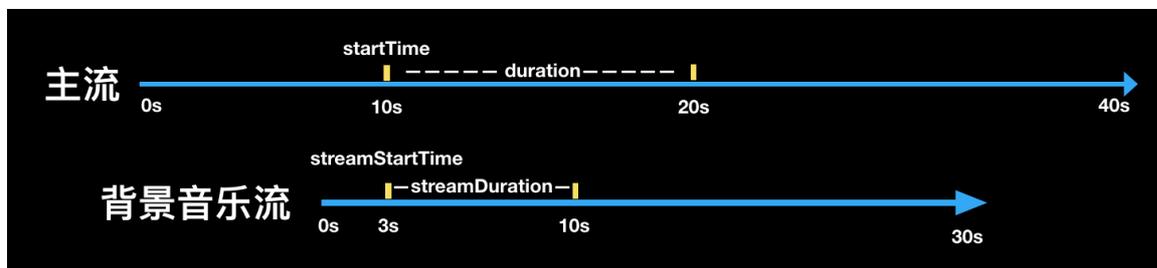
编辑时添加特效，使用AliyunICompose.compose合成出来的视频不带特效，是什么原因？

V3.5.0及以前的版本，要把编辑预览的特效持久化到本地的配置文件中，需要调用AliyunEditor.onPause接口，如果没有调用，则特效配置不会持久化到本地文件中，那么通过AliyunICompose接口反序列化生成的Project就不带特效，也就导致合成出来的视频不带特效。

编辑时，调用applyMusic。添加音乐后，设置了startTime，为什么每次音乐流都从0开始播，而不是从startTime开始？

查看接口文档EffectBean类里对startTime的解释，startTime指的是特效在主流上的作用时间，并不是指素材流的起始时间，在V3.6.0版本之后增加了一个streamStartTime参数，这个参数是表示素材流的起始时间，即有这种需求的开发者，在V3.6.0及以后的版本中可以通过接口参数配置实现，但是V3.6.0之前的版本，只能由开发者先对素材流做裁剪，然后用裁剪的音乐去做背景音乐。

如下图所示：在播放到10s的时候，背景音乐开始播放，从音乐流的3s处播放，播放到10s处，重新从3s处开始播放，当第二遍播放到音乐流的6s时，刚好对应主流的20s位置，停止播放背景音乐。



如何添加gif作为主体流？

V3.7.0以下版本，gif格式的文件是作为图片类型导入；V3.7.0及以上版本，gif格式的文件如果作为视频类型导入，则当做视频播放gif所有帧，如果作为图片类型导入，则作为图片播放第一帧。

为什么设置完转场或者applySourceChange之后，视频卡住不动了？

需要您在进行这些操作之后调用mAliyunEditor.play()接口。

4.2. iOS常见问题

本文主要介绍短视频SDK在iOS端的使用问题。

SDK集成

导入短视频SDK时，控制台提示category方法未找到，该如何解决？

在工程target中选择Build Setting > other linker flags，添加-ObjC。

Debug包和Release包有什么区别？

Debug包包包含模拟器和真机版本，可以保证模拟器编译通过。在提交App Store时使用Release包，因为Apple要求动态库提交不能包含模拟器版本。

导入短视频SDK后运行crash，提示image not found，该如何解决？

短视频SDK使用了动态库，导入动态库需要在Embedded Binaries中添加对应的framework，详细操作请参考[工程配置](#)。

短视频SDK支持bitcode吗？

短视频SDK不支持bitcode，需要在设置中把Enable Bitcode设为NO。打包如果出现failed to verify bitcode错误，需要取消勾选rebuild for bitcode选项。

提示[NSDictionary oss_dictionaryWithXMLData:]: unrecognized selector sent to class, 是什么原因?

未导入上传SDK, 短视频SDK需要依赖上传SDK的 AliyunOSSiOS.framework。

视频录制

普通录制完成后, 获取不到视频, 该如何解决?

需要在录制完成回调函数 `-(void)recorderDidFinishRecording;` 里才能获取到视频。

如何实现横屏录制?

录制时候设置 `cameraRotate` 角度值, 录制的视频方向会以第一段视频的角度值为准。

录制过程中更换音乐, 没有生效, 是什么原因?

录制过程中不支持更换音乐。

如何实现全屏录制方案?

录制分辨率9: 16显示有以下两种方案:

- 方案一: 和短视频SDK提供的demo中一致, iphone x上下留黑边。
- 方案二: 可以调整view布局上下撑满, 左右一部分内容不显示。

视频裁剪

裁剪提示1008错误, 该如何解决?

关闭 `shouldOptimize` 选项。

裁剪提示700004错误, 该如何解决?

输出路径未设置, 请设置输出路径。

如何实现没有黑边的裁剪?

根据原始分辨率, 做一个缩放, 缩放后的分辨率保证是偶数。

如何裁剪一段音乐?

裁剪参数 `videoSize` 和 `ouputSize` 都无需设置, 其他操作和裁剪视频时的参数保持一致。

视频编辑

编辑完成后, 合成crash, 出现报错提示[null length], 该如何解决?

检查水印路径是否设置正确。

调用音量接口出现破音, 是什么原因?

音量默认值100代表原声, 大于100可能会破音, 建议保证音量值设置在0~100。

滤镜、MV等资源找不到, 该如何解决?

资源拷贝到项目中需要用folder方式导入才能保证层级关系, 注意在xcode中显示的文件夹是蓝色的。

导入视频, 提示operation not permit, 该如何解决?

从系统相册导入的视频, 需要调用系统接口获取相册访问权限, 同时保证对应的AVAsset没有被销毁。

加入音频后, 无法调节音量和添加、删除音效, 该如何解决?

注意加入的音频需要为pcm、mp3等格式的音频文件, 不能为视频文件。

视频上传

上传报错，提示没有授权，该如何解决？

获取STS流程简单介绍：

1. 使用阿里云主账号创建用户，给用户授予AliyunSTSAssumeRoleAccess权限。
2. 使用用户创建角色，给角色授予VODFULL权限。
3. 通过调用STS的SDK，获取STS，参考：[Java示例](#)。

 说明 调用STS的SDK中的ak必须要是子账户的ak。

4. 修改policy为点播的VODFULL权限。

```
String policy = "{\n" +
  "  \"Version\": \"1\", \n" +
  "  \"Statement\": [\n" +
  "    {\n" +
  "      \"Action\": [\n" +
  "        \"vod:*\"\n" +
  "      ], \n" +
  "      \"Resource\": [\n" +
  "        \"*\"\n" +
  "      ], \n" +
  "      \"Effect\": \"Allow\"\n" +
  "    }\n" +
  "  ]\n" +
  "}"
```

上传过程中断网，为什么没有失败回调？

上传过程中断网，会自动重试，如果不想走重试接口，可以手动调用取消上传的接口。

上传后的视频，通过服务端SDK下载的视频格式为什么是m3u8？

转码配置里面，如果勾选了hls选项，则会生成m3u8格式的视频。

4.3. SDK和License常见问题

本文主要介绍使用短视频SDK时遇到的SDK使用、试用、购买及License等问题。

如何申请试用短视频SDK？

请发送公司名称、应用名称、申请试用的SDK版本（基础、标准、专业）、联系人、联系电话、应用bundleID、包名和签名信息（MD5格式小写无冒号）、阿里云的账号或UID（若没有账号请注册）至邮箱videosdk@service.aliyun.com，申请开通试用（以上资料请保证齐全、格式规范），或直接联系商务经理申请开通。2个工作日之内处理完开通申请。未开通之前，如着急试用，可先自行集成测试（最多可用7天）。

试用期默认为一个月，如需延长试用期可，可再次申请试用。

如何免费使用短视频SDK？

目前购买点播套餐即可免费使用短视频SDK，详细请参见[获赠License授权](#)。

如何购买专业版短视频SDK？

发送试用邮件之后会有商务经理同您沟通。具体专业版和其他版本的区别，请参见：[基础版、标准版、专业版功能比较](#)。

提示License Failed，该怎么办？

1. 确保已经购买点播套餐或者发送申请试用邮件。
2. iOS确保bundleID和申请试用的bundleid是一致的。
3. Android确保包名和签名（MD5）获取是否一致。这里提供一个[签名获取工具](#)，安装后按工具中的指导操作。

人脸贴纸与人脸AR、动态贴纸的区别？

人脸贴纸属于视频录制中的功能，即通过人脸识别的功能，给识别到的人脸添加指定的贴纸，目前仅专业版本支持。人脸AR属于美颜特效SDK中的功能，需要额外授权。动态贴纸属于视频编辑中的功能，仅专业版本支持。

短视频SDK Demo中部分资源的获取地址？

- 人脸贴纸资源下载地址：[人脸贴纸下载](#)
- 视频编辑时的动图资源下载地址：[动图下载1](#)及[动图下载2](#)

5.播放器问题

5.1. Web播放器常见问题

Aliplayer播放器包含H5、Flash、自适应播放器，建议用户选择自适应播放器，可以根据终端类型、浏览器类型和地址协议选择最合适的播放器。视频加密播放可以使用H5播放器。

如何手工启用H5播放器

手工启用H5播放器有两种方式：

- 直接引用H5播放器的js文件。
- 使用自适应播放器，然后设置useH5Prism属性为true。

如何手工启用Flash播放器

手工启用Flash播放器有两种方式：

- 直接引用Flash播放器的js文件。
- 使用自适应播放器，然后设置useFlashPrism属性为true。

如何自适应播放器

根据终端类型、浏览器类型、设置的属性和地址协议选择最合适的播放器，适配的基本原则是：

- H5优先级最高，能H5播放的绝不选择Flash，除非用户指定用Flash播放。
useFlashPrism = true、rtmp和http-flv协议时，采用flash播放。
- 移动端采用H5播放。
useH5Prism = true，采用H5播放。
- PC端MP4采用H5播放。
PC端如果浏览器或通过Aliplayer的插件支持播放m3u8，则采用H5播放，否则采用Flash播放。
- 其它都用H5播放。

哪些浏览器支持flash

播放应该都支持flash播放，但是最新的一些浏览器会禁用flash，需要手工启用，参见以下说明：

- [IE使用说明](#)
- [Firefox使用说明](#)
- [Chrome使用说明](#)

flash播放器对mp4/flv无法拖拽

mp4与flv拖拽需要CDN添加支持，是通过播放器发送带时间的请求到CDN，CDN返回该时间段的视频数据。如果要实现拖拽，需要以下两个条件：

- 文件索引信息需要在视频的头部，mp4包含视频时间戳等索引信息，以及flv的meta信息要在视频最前面，播放器解析到视频索引信息后，才可以依据拖拽的位置通过索引信息拿到指定位置的数据点，去向CDN发送请求。
- CDN支持带时间/byte range的请求，需要在CDN控制台开启，如果在控制台开启，请参见[配置拖拽播放](#)。

解决Andorid微信上自动弹出全屏播放

Android手机在微信和QQ浏览器里自动全屏播放，这是腾讯浏览器的内置行为，不能修改，原因是由于腾讯浏览器挟持了video标签，由腾讯内置的播放器播放视频，但可以启用同层播放功能，可以解决视频覆盖Dom元素的问题[如何启用H5的同层播放](#)。

在微信里如何自动播放

```
<script src="http://res.wx.qq.com/open/js/jweixin-1.0.0.js"></script>
<script>
function autoPlay() {
wx.config({
  // 配置信息,即使不正确也能使用 wx.ready
  debug: false,
  appId: '',
  timestamp: 1,
  nonceStr: '',
  signature: '',
  jsApiList: []
});
wx.ready(function() {
  var video=$(player.el()).find('video')[0];
  video.play();
});
};
// 解决ios不自动播放的问题
autoPlay();
</script>
```

播放器如何初始播放位置

H5播放器：

```
var seeked = false;
player.on('canplaythrough',function (e) {
if(!seeked)
{
  seeked = true;
  player.seek(100);
}
});
```

Flash 播放器：

```
var seeked = false;
player.on('loadedmetadata',function (e) {
if(!seeked)
{
  seeked = true;
  player.seek(20);
}
});
```

如何禁用进度条

自定义skinLayout属性，去掉整个controlBar或者controlBar下面的子项，比如progress：

```
skinLayout: [
  {name: "bigPlayButton", align: "blabs", x: 30, y: 80},
  {
    name: "H5Loading", align: "cc"
  },
  {
    name: "controlBar", align: "blabs", x: 0, y: 0,
    children: [
      //{name: "progress", align: "tlabs", x: 0, y: 0},
      {name: "playButton", align: "tl", x: 15, y: 26},
      {name: "timeDisplay", align: "tl", x: 10, y: 24},
      {name: "fullScreenButton", align: "tr", x: 20, y: 25},
      {name: "volume", align: "tr", x: 20, y: 25},
    ]
  }
]
```

如何切换vid和playauth

H5播放器，直接调用replayByVidAndPlayauth方法：

```
player.replayByVidAndPlayauth(newVid, newPlayAuth)
```

Flash播放器需要销毁，根据新的vid和playauth重新创建一个：

```
//销毁
flashPlayer.dispose();
$('#flashPlayer').empty();
//重新创建
flashPlayer = new Aliplayer({
  id: 'flashPlayer',
  autoplay: true,
  playsinline: true,
  vid: newVid,
  playauth: newPlayAuth,
  useFlashPrism: true
});
```

如何定时获取播放时间

通过定时器每秒调用播放器的getCurrentTime方法获取播放时间，在暂停、出错和结束播放时清除定时器。

```
var timer = null;
function getTime(){
    var currentTime = player.getCurrentTime();
    //to do
    timer = setTimeout(getTime,1000);
}
//清除定时器
function clear(){
    if(timer)
    {
        clearTimeout(timer);
        timer = null;
    }
}
player.on('ended',function (e) {
    clear();
});
player.on('pause',function (e) {
    clear();
});
player.on('error',function (e) {
    clear();
});
```

如何调整H5播放器的播放按钮的大小和位置

重写CSS，比如减小一倍：

```
.prism-player .prism-big-play-btn {
    width: 45px;
    height: 45px;
    background-size: 128px 256px;
}
```

位置可以通过设置skinLayout里bigPlayButton的x、y属性：

```
skinLayout: [
    {name: "bigPlayButton", align: "blabs", x: 30, y: 80},
    {
        name: "H5Loading", align: "cc"
    },
    {
        name: "controlBar", align: "blabs", x: 0, y: 0,
        children: [
            {name: "progress", align: "tlabs", x: 0, y: 0},
            {name: "playButton", align: "tl", x: 15, y: 26},
            {name: "timeDisplay", align: "tl", x: 10, y: 24},
            {name: "fullScreenButton", align: "tr", x: 20, y: 25},
            {name: "volume", align: "tr", x: 20, y: 25},
        ]
    }
]
```

手机端播放视频不希望弹出全屏，要小窗播放问题

手机端不希望全屏播放，iOS可以设置属性playsinline为true。

Android手机在微信和QQ浏览器里自动全屏播放，这是腾讯浏览器的内置行为，不能修改，原因是由于腾讯浏览器挟持了video标签，由腾讯内置的播放器播放视频，但可以启用同层播放功能，可以解决视频覆盖Dom元素的问题[如何启用H5的同层播放](#)。

启用IE浏览器以最高级别的可用模式显示内容

小于IE10的浏览器需要启用最高级别的可用模式显示内容模式：

```
<meta http-equiv="x-ua-compatible" content="IE=edge" >
```

Flash播放器播放m3u8提示跨域错误

播放器跨域访问时需要添加策略文件，即在视频播放链接所在域名的根目录下，添加crossdomain.xml文件，其中添加播放器所在域名的权限。

例如：<http://example.com/app/test.m3u8> 需要添加到 <http://example.com/crossdomain.xml> 。

```
<?xml version="1.0" encoding="UTF-8"?>
<cross-domain-policy>
  <allow-access-from domain="*" />
  <allow-http-request-headers-from domain="*" headers="*" secure="false" />
</cross-domain-policy>
```

Flash播放器封面图片无法显示

- 确认cover字段输入URL是否有效。
- 确认cover输入的URL所在域名是否存在有效的crossdomain.xml文件。

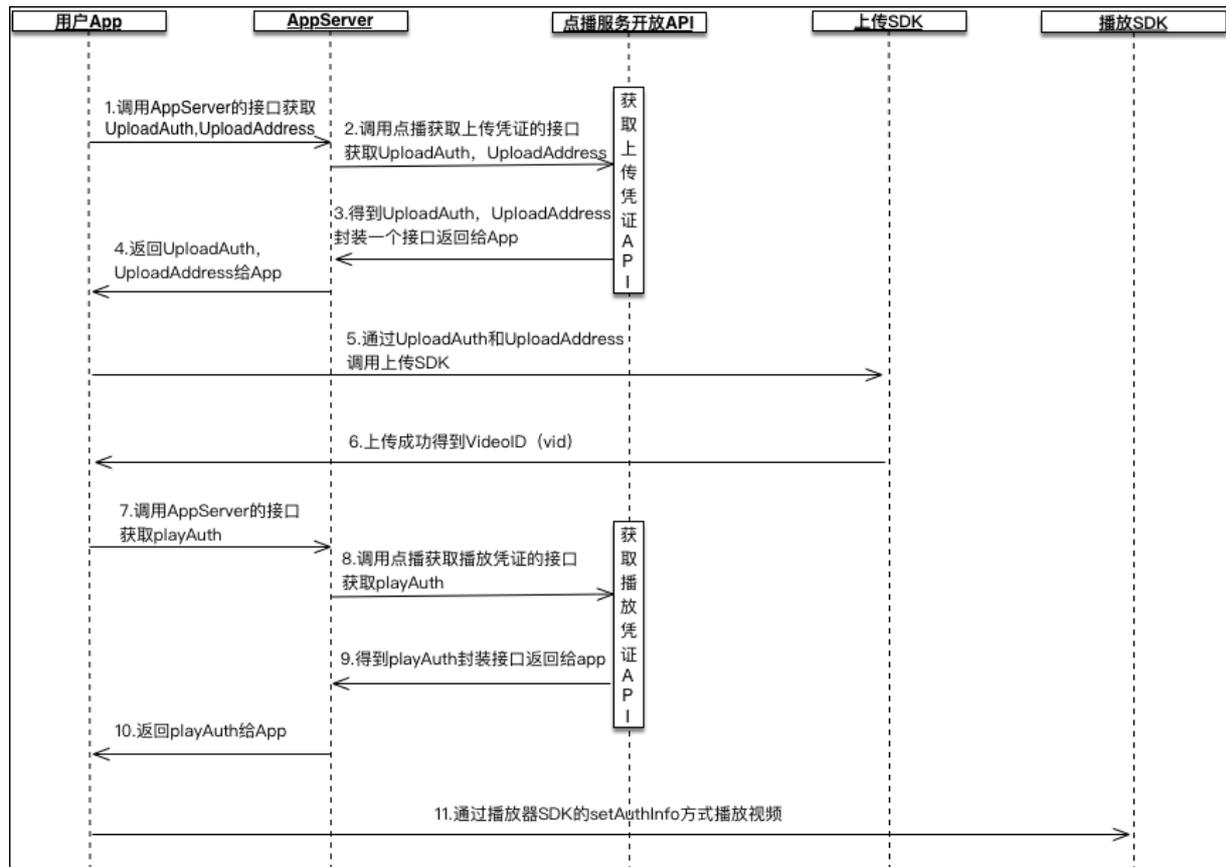
5.2. 播放器调试和实现流程问题

播放器为免费产品，建议搭配视频点播、视频直播、媒体处理等产品使用，以便获得更优体验，比如清晰度切换、安全播放、加密下载等。

如何实现客户端上传并播放？

媒体上传流程解析，请参见[概述](#)。

实现流程：用户App获取上传凭证 -> 服务端下发上传凭证 -> 用户上传视频并获取vid -> 服务端获取播放凭证 -> 将播放凭证下发给客户端 -> 完成视频播放，具体见下图。

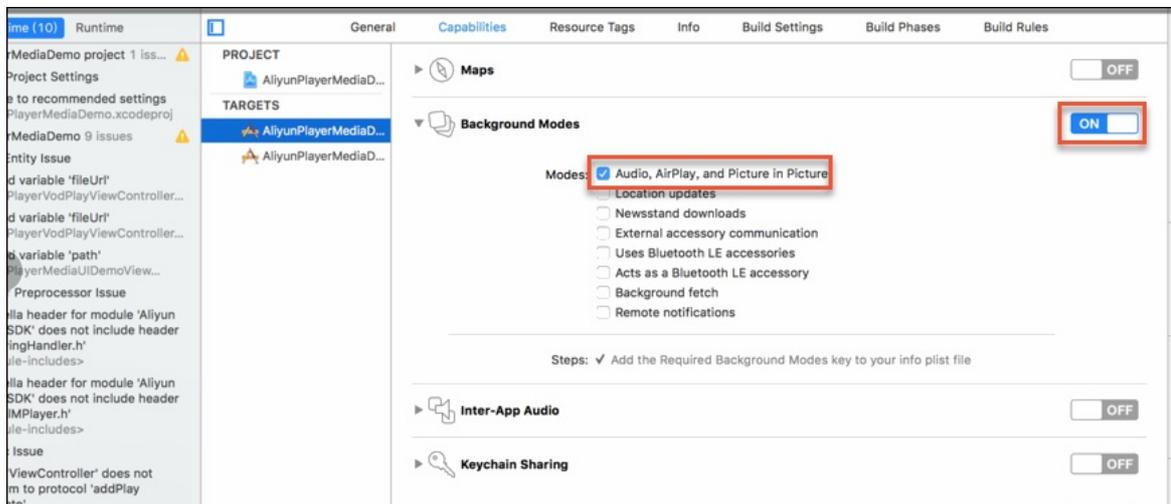


说明 目前视频上传到视频点播平台，并且转码成功后才可进行播放。需要通过视频转码成功回调来知晓视频转码是否完成，请参见[视频转码完成](#)。

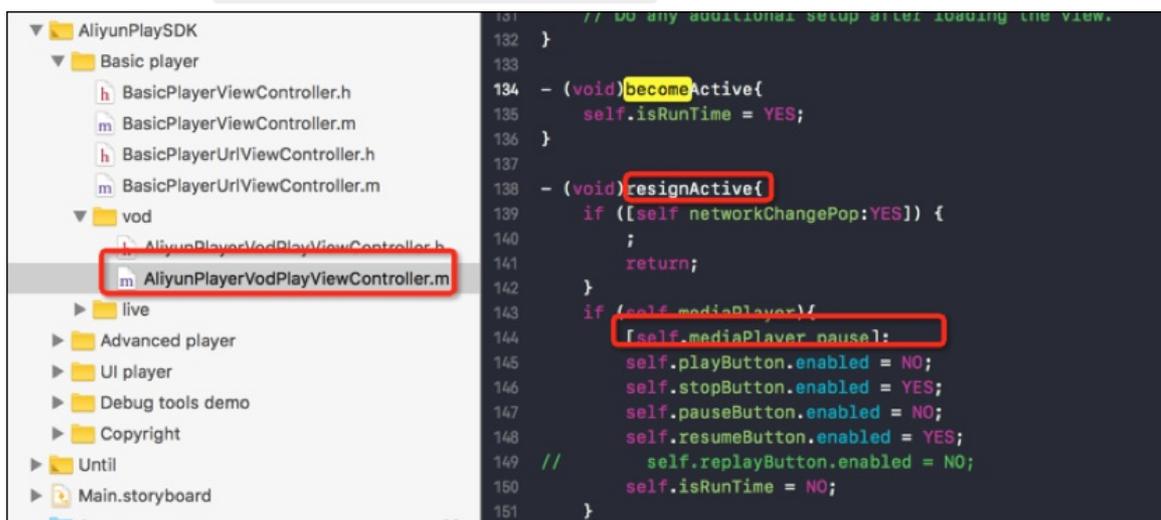
iOS播放器无法在后台播放视频

iOS播放器SDK无法支持后台播放视频，即按home键以后还能播放，用Demo测试也无法播放。解决办法如下：

- 1. 设置Background Modes，在xcode里开启后台采集数据功能。



2. 在vod目录下找到 AliyunPlayerVodPlayViewController.m 文件。



3. 在该文件的 becomeActive方法下面有个对应 resignActive方法。resignActive方法里面带有 pause方法，将 pause方法注释掉，如下图所示。



各版本更新了哪些内容？

以播放器SDK为例，请参见[播放器发布历史](#)。

客户端开发快速获取playAuth进行播放器调试？

客户端开发可以通过python脚本获取播放凭证进行调试，具体流程如下：

1. 安装python2.7+pip环境（Mac环境下自带，Windows环境自行安装）。
2. 使用终端安装SDK，运行以下命令：

```
pip install aliyun-python-sdk-core
pip install aliyun-python-sdk-vod
```

3. 下载Python脚本，从阿里云控制台获取accessKeyId和accessKeySecret并替换脚本里面的字段内容，从点播控制台获取已经上传视频的videoId并替换脚本里面的字段内容。
4. 在python脚本所在目录下通过终端执行以下命令：

```
python playAuth.py
```

5. 在终端中查看获取的PlayAuth和VideoId。
6. 在播放器SDK中使用获取的PlayAuth和VideoId进行播放。

🔍 说明 调用此方式时，需注意PlayAuth的时效问题。因为PlayAuth只有100s的时效。多数播放不了的问题是由此引起。

5.3. 安全文件获取问题

出于安全性的考虑，点播支持对视频进行加密，视频加密之后使用播放器进行解密，这就需要一个加解密的流程，因此播放器下载就需要先获取一个安全文件。本文为您介绍获取安全文件的操作步骤。

适用人群

对于客户端播放器实现下载需要的安全文件的开发者。

开启安全下载并生成安全文件

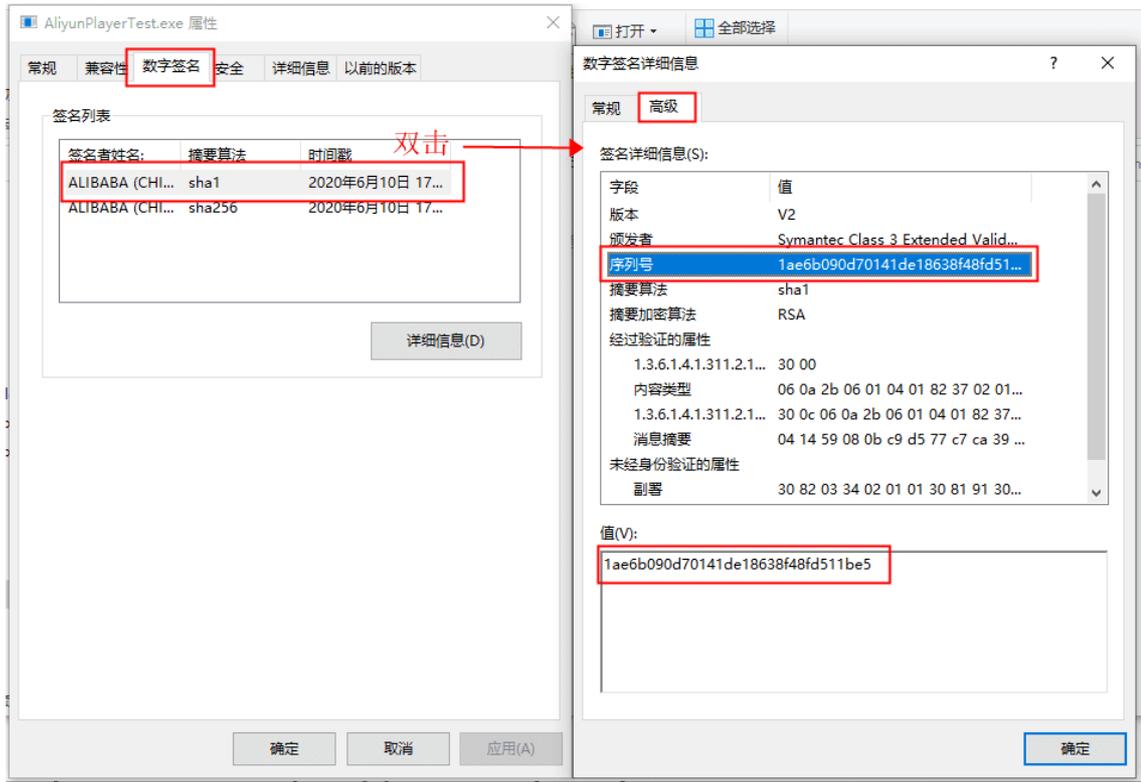
1. 保证已开启点播服务。如何开启阿里云点播服务，具体操作，请参见[开通视频点播](#)。
2. 在点播控制台开启安全下载功能，具体操作，请参见[下载设置](#)。
3. App唯一标识获取。
 - 安卓：需要提供一个SHA1的密钥。可以使用以下代码段直接生成密钥。

```
//这个是获取SHA1的方法
public static String getCertificateSHA1Fingerprint(Context context) {
    //获取包管理器
    PackageManager pm = context.getPackageManager();
    //获取当前要获取SHA1值的包名，也可以用其他的包名，但需要注意
    //在用其他包名的前提是，此方法传递的参数Context应该是对应包的上下文。
    String packageName = context.getPackageName();
    //返回包括在包中的签名信息
    int flags = PackageManager.GET_SIGNATURES;
    PackageInfo packageInfo = null;
    try {
        //获得包的所有内容信息类
        packageInfo = pm.getPackageInfo(packageName, flags);
    } catch (PackageManager.NameNotFoundException e) {
        e.printStackTrace();
    }
    //签名信息
    Signature[] signatures = packageInfo.signatures;
    byte[] cert = signatures[0].toByteArray();
    //将签名转换为字节数组流
    InputStream inout = new ByteArrayInputStream(cert);
}
```

```
...private CertificateFactory cf = null;
//证书工厂类，这个类实现了出厂合格证算法的功能
CertificateFactory cf = null;
try {
    cf = CertificateFactory.getInstance("X509");
} catch (CertificateException e) {
    e.printStackTrace();
}
//X509证书，X.509是一种非常通用的证书格式
X509Certificate c = null;
try {
    c = (X509Certificate) cf.generateCertificate(input);
} catch (CertificateException e) {
    e.printStackTrace();
}
String hexString = null;
try {
    //加密算法的类，这里的参数可以使用MD4,MD5等加密算法
    MessageDigest md = MessageDigest.getInstance("SHA1");
    //获得公钥
    byte[] publicKey = md.digest(c.getEncoded());
    //字节到十六进制的格式转换
    hexString = byte2HexFormatted(publicKey);
} catch (NoSuchAlgorithmException e1) {
    e1.printStackTrace();
} catch (CertificateEncodingException e) {
    e.printStackTrace();
}
return hexString;
}
//这里是将获取到的编码进行16进制转换
private static String byte2HexFormatted(byte[] arr){
    StringBuilder str = new StringBuilder(arr.length * 2);
    for (int i = 0; i < arr.length; i++)
    {
        String h = Integer.toHexString(arr[i]);
        int l = h.length();
        if (l == 1)
            h = "0" + h;
        if (l > 2)
            h = h.substring(l - 2, l);
        str.append(h.toUpperCase());
        if (i < (arr.length - 1))
            str.append(':');
    }
    return str.toString();}
}
```

- o iOS: 直接填bundleID即可

- Windows: 调用SDK的exe的数字签名证书的序列号, 如下图所示。



- 填写离线解密私钥。填写16~32位的字符串, 同时包含大小写字母和数字。
- 填写完成之后单击生成密钥并下载, 就可以获取安全图片。

注意 提供获取SHA1的算法, 前提需要获取默认的keystore签名的SHA1。如果你最终的release包是另外的keystore, 建议直接在gradle文件中直接配置release keystore, 然后通过上述方法获取SHA1。

配置keystore示例代码。

//这里演示如何配置keystore,下面的keystore文件开发者可以自由替换,一下配置的前提是开发者将keystore文件放在项目的根目录下。

```
signingConfigs {
    debug {
        storeFile file("$rootDir/debug.keystore")
        storePassword "android"
        keyAlias "androiddebugkey"
        keyPassword "android"
    }
    release {
        storeFile file("$rootDir/debug.keystore")
        storePassword "android"
        keyAlias "androiddebugkey"
        keyPassword "android"
    }
}
buildTypes {
    debug {
        multiDexEnabled true
        signingConfig signingConfigs.debug
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
    release {
        minifyEnabled true
        multiDexEnabled true
        signingConfig signingConfigs.release
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}
```

5.4. 参数解析问题

您可以通过阅读本文了解视频点播中部分重要参数的解释以及说明。

videoID是什么？为什么需要videoID？如何获取？

出于安全性考虑，用户使用点播上传媒体文件，将获得该文件的视频ID（videoID），而非URL地址。用户也可以通过点播OpenAPI获取，更多内容，请参见：[获取视频播放地址](#)。

使用点播上传完成之后，会获取到一个videoID。

videoID也可以在阿里云点播控制台获取，操作流程如下所示：

1. 登录[视频点播控制台](#)。
2. 在点播控制台左侧导航栏的媒资库区域，单击音/视频。
3. 在视频列表内获取videoID（视频ID）。

可以通过控制台获取的videoID来做下载和播放的测试。如何实现点播上传见：[概述](#)。

AccessKey Id和AccessKey Secret是什么？如何获取？

阿里云AccessKey Id和AccessKey Secret是您访问阿里云API的唯一凭证。AccessKey Id是类似身份的标识，而AccessKey Secret的作用是签名您的访问参数，以防被篡改。AccessKey Secret类似您的登录密码，不要向任何人泄露。

获取操作：

1. 登录[视频点播控制台](#)。
2. 将鼠标悬浮在上方菜单栏的用户区域，在弹出的快捷菜单中单击AccessKey管理。
3. 在AccessKey管理页面，创建AccessKey，或者查看已有AccessKey Id的AccessKey Secret。

playKey是什么？如何获取？

playkey (apikey)：播放密钥，用于播放器SDK获取视频播放地址时验证身份，播放鉴权是视频点播在阿里云AK安全认证基础上的二次鉴权机制，可有效防止盗链。根据用户播放时可能使用的平台，默认提供Flash、H5、iOS、Android四个平台的播放密钥。

 **说明** 为保证密钥安全，查看播放密钥时需要输入手机号验证码确认身份。

获取操作：

1. 登录[视频点播控制台](#)。
2. 在点播控制台左侧导航栏的**配置管理**区域，选择**分发加速配置 > 下载设置**。开启安全下载模式。
3. 在获取密钥中，填写APP唯一标识和离线解密私钥。
4. 单击生成密钥并下载。

playauth是什么？如何获取？

目前播放器播放视频分为三种模式，对应三种不同的使用场景。使用playauth的方式最安全，建议使用setAuthInfo的播放模式来实现。

playauth可以理解为点播服务将所有的信息（videoID、AccessKey Id、AccessKey Secret）都进行混合再做加密操作，用户拿到的就是一串包含多个信息的数据，播放器就可以播放。

播放模式	适用场景	优劣	是否建议使用
setDataSource	适用于测试	有泄漏风险，需要将自己的AccessKey Id和AccessKey Secret直接写入客户端代码中，客户端如果被破解将有泄露风险。	不建议商用时上线使用
setAuthInfo	适用于正式商用	安全，所有的视频地址和链接都不暴露。	建议商用使用
播放本地和网络URL	可以播放本地视频且能够播放视频URL	简单，可以播放其他平台的视频。	在有播放本地视频需求和播放网络视频需求时使用

流程：**服务端获取播放凭证 > 将播放凭证下发给客户端 > 完成视频播放。**

1. 获取播放凭证：用户在服务器侧通过调用播放鉴权SDK（Server端的SDK），向视频点播服务获取播放凭证。
2. 完成视频播放：播放器SDK根据视频ID和播放凭证向视频点播服务获取视频的播放地址，从而加载视频流并解码完成播放。

注意

播放凭证时效为100秒，只能用于获取指定视频的播放地址，不能混用或重复使用。如果凭证过期则无法获取播放地址，请重新获取凭证。

播放器SDK根据播放凭证会自动获取播放地址进行解码播放，播放地址的时效为30分钟，若播放凭证失效请重新获取播放凭证，并回传给播放器SDK用于刷新播放地址。

为保障主账号安全，建议使用子账号的AccessKey，尤其在Web播放场景下。

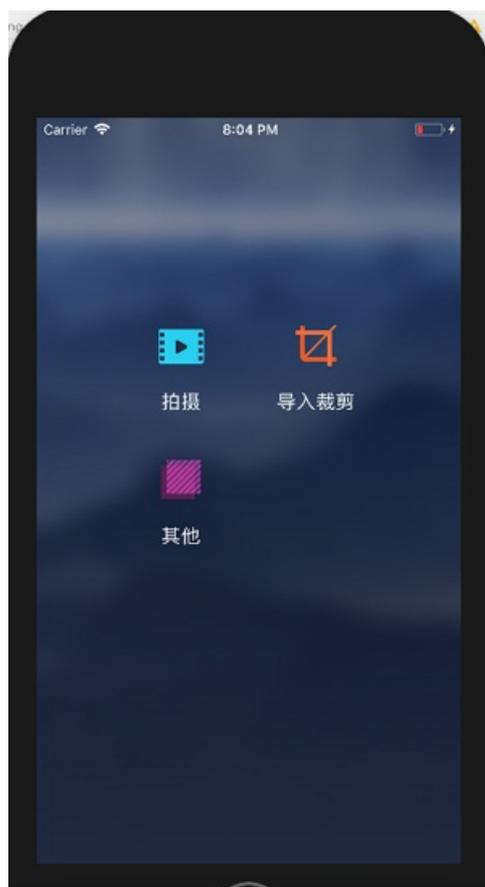
5.5. 短视频SDK在IOS模拟器上安装和使用的方法

详细信息

1. 下载官网的短视频Demo和短视频SDK AAR包。
2. 包解压之后，将SDK目录下Debug中的包复制，替换掉Demo中对应的同名包，具体如下所示。



3. 完毕之后，使用Xcode接入，接入结果如下所示。



6. 事件通知常见问题

本文为您介绍事件通知常见问题。

没有收到HTTP回调该如何排查？

- 首先需要登录[视频点播控制台](#)，在点播控制台左侧导航栏的**配置管理**，选择**媒体处理配置 > 回调设置**，确认是否开启回调功能，并且查看是否订阅了所需要的回调。
- 执行下述示例，查看是否有正常返回，如果没有，则排查回调地址接收服务是否有问题。注意将 `http://example.aliyundoc.com/processMessage` 替换为您设置的HTTP回调地址。

```
curl -l -i -H "Content-type: application/json" -X POST -d '{"VidoeId":"videoid","EventType":"FileUploadComplete","Status":"success","Size":1439213}' http://example.aliyundoc.com/processMessage
```

回调超时时间是多少？重试几次？重试间隔多久？

- HTTP回调超时时间默认为5秒，最多重试3次，每次间隔1秒。

超过重试次数后，消息会被丢弃吗？这样如何保证消息不丢失？

- 如果您的接收服务异常，如崩溃、重启、网络不可达等，超过重试次数消息都会被丢弃。如果要保证消息不丢失，推荐使用MNS回调。

支持302跳转么？

- HTTP回调只支持200响应，出于安全考虑，不支持302、301等。

收到多次重复回调是什么原因？

- HTTP回调只有收到200响应码时才认为是成功，否则均认为失败，并重试回调，最多尝试三次。

HTTP鉴权有什么作用？

- HTTP鉴权是在回调过程增加特殊HTTP头，供回调地址提供方进行校验（防止非阿里云点播服务调用该接口，造成安全问题）。具体是否校验可由用户自行判断。

MNS回调

队列中没有消息是什么原因

- 未授权点播服务访问、提交的工单中Endpoint不是公网或者队列名称不对会导致消息投递失败。
- 消息存活时间设置过小（建议使用3600秒），消息被消费前已经释放。
- 消息最大长度设置过小（建议使用默认值65536 Byte），可能会导致消息投递失败。

收到重复消息是什么原因

- 消息被消费后，变为短暂不可见，需要手动或调用接口删除消息，否则一段时间后消息可以被再次消费。

MNS回调是否支持鉴权

- 点播服务经过授权写入用户队列，非经授权无操作权限，安全性比较高，所以一般不需要鉴权。

回调支持哪些区域

- 如果视频保存在**中国大陆区域**存储（如**华北2**，**华东2**），投递消息到**华东2（上海）**区域之外的队列

存在较短时间的延迟，建议使用华东2（上海）区域的队列。

- 如果视频保存在新加坡区域存储，建议使用新加坡区域的消息队列。
- 如果视频保存在法兰克福区域存储，建议使用法兰克福区域的消息队列。

是否能跨区域配置MNS队列进行回调？

- 是否能跨区域配置MNS队列进行回调，比如视频存储和处理在华东2（上海）或华北2（北京），但MNS队列在华南1（深圳）。功能上是支持的，只是要考虑下网络延迟，可能消息写入会稍慢一些，建议尽量配置和存储地址在同一区域，如华东2。

7.如何启用H5的同层播放

您可以阅读本文，了解在H5的同层播放功能。

背景信息

阿里云播放器2.0.1及以上，支持在Android微信小程序上支持H5同层播放功能，此功能只对Android腾讯的X5浏览器起作用。

如果微信没有启用同层播放，在Android微信上播放视频时，会自动弹出全屏播放，覆盖Dom元素。

同层播放属性

名称	类型	说明
x5_type	String	启用同层播放，设置值为h5。
x5_fullscreen	Boolean	声明视频播放时是否进入到TBS的全屏模式，支持的值为true。
x5_video_position	String	声明视频播在界面上的位置，默认为center。取值： <ul style="list-style-type: none"> • top：顶部。 • center：中央。
x5_orientation	String	声明TBS播放器支持的方向，取值： <ul style="list-style-type: none"> • landscape：横屏。 • portrait：竖屏。

同层播放设置

• 不全屏同层播放

通过设置x5_type属性为h5，将playsinline设置为false，启用同层播放。

通过x5_video_position定义视频的位置，如果在顶部可以设置为top，居中可以设置center，默认为center。

Demo，请下载[H5 Demo](#)。

• 全屏同层播放

通过设置x5_type属性为h5，将playsinline设置为false，启用同层播放，设置x5_fullscreen为true，启用全屏，全屏播放器不需要设置x5_video_position属性。

全屏播放视频默认是平铺的，如果想不平铺可以设置object-fit的样式为contain或其它。

```
video {
  object-fit: contain !important;
}
```

Demo，请下载[H5 Live Demo](#)。

同层播放的建议

在使用同层播放器时，为您提供的一些建议，如下所示：

- 监听resize事件实现自适应视口大小变化，视频播放时会调整视口大小。
- 在视频播放期间的交互、弹框和字幕要在视频区域中，不要在视频区域外。
- 对于直播类全屏视频，最好不要在最顶部放交互性元素。
- 对于需要全屏交互的，可以将video区域设置为视口大小。

更多设置

- 进入和退出同层播放时布局处理

在进入和退出同层播放的事件里，可以添加一些调整布局的逻辑，时常在进入同层播放时，布局需要做一下调整。例如全屏，元素的位置等。

进入事件：x5requestFullScreen。

退出事件：x5cancelFullScreen。

- 更多的自定义

如果x_video_position属性的top和center不能满足要求，可以通过自定义object-fit和object-position属性，进行更灵活的设置视频的显示位置。

- object-fit属性

object-fit：CSS属性指定替换元素的内容应该如何适应到其使用的高度和宽度确定的框。

取值：

- object-fit：fill。
- object-fit：contain
- object-fit：cover
- object-fit：none
- object-fit：scale-down

CSS代码如下所示：

```
video {  
  object-fit: contain !important;  
}
```

- object-position属性

object-position：CSS属性控制替换内容位置，和background-position属性很类似。

CSS代码如下所示：

```
img {
  width: 300px;
  height: 250px;
  border: 1px solid black;
  background-color: silver;
  margin-right: 1em;
  object-fit: none;
}
#object-position-1 {
  object-position: 10px;
}
#object-position-2 {
  object-position: 100% 10%;
}
```

- 调整播放器容器的高度

由于设置了视频的位置，会引起controlbar显示不是在视频的最下面，可以通过订阅resize和requestFullScreen事件调整视频容器的高度。

```
var setLayout = function()
{
  //设置播放器容器的高度
  var height ;//根据实际情况设置高度
  player.el().style.height = height;
}
window.onresize = function(){
  setLayout();
}
player.on("requestFullScreen", function(){
  setLayout();
});
```

8.HLS标准加密常见问题

如何使用生成的密钥？

GenerateDataKey接口会返回两种密钥：一种是密文密钥（CipherText），另一种是明文密钥（PlainText）。用户只需要将密文密钥传递给视频点播服务即可，具体参数传递请参见[提交媒体转码作业中的加密配置 EncryptConfig](#)参数。

说明

- 推荐业务方对生成的密文密钥和明文密钥进行缓存。
- 创建的Service Key不可删除、不可更新，只用于生成加密密钥。
- 关于密钥相关费用，参见[密钥管理服务-计费方式-API调用费用](#)。

生成的令牌如何传递到解密接口？

业务方必须使用域名进行加速播放，才能将令牌重写到解密接口上。在请求M3U8地址时需要将生成的令牌通过MtSlsUriToken参数传递，域名会自动将该参数重写到解密接口上并请求解密接口。

如何快速验证加密播放？

可以通过[阿里云播放器诊断平台](#)，播放标准加密的M3U8播放地址，可以快速验证播放是否正常。拷贝M3U8播放地址（如有MtSlsUriToken参数一并拷贝）到阿里云播放器诊断平台上进行尝试标准加密的解密播放。

其它常见问题

- 接口提示
调用[提交媒体转码作业](#)提示KeyNotFound相关信息，请先联系点播后台在相应的区域（例如：华北2、华东2）创建Service Key并用于生成加密密钥。
- 非加密文件
生成的文件未加密，请确认转码模板是否开启[视频加密](#)并勾选了[私有加密](#)。
- 自定义密钥
加密的明文密钥必须使用[GenerateDataKey](#)生成，不能使用自定义字符串生成加密密钥，否则加密转码失败。
- 加密失败
标准加密转码失败或没有任何加密文件生成，请确认[GenerateDataKey](#)生成的密钥类型是否为AES_128密钥，否则加密转码会失败导致不会生成任何加密文件。
- 解密失败
标准加密文件解密播放失败，请确认是解密接口应该将KMS生成的明文密钥再次base64decode之后给播放器解密播放，不然解密播放会失败。
- 重复生成
标准加密转码触发都是由用户主动触发，如果重复生成加密文件，请确认是否重复调用[提交媒体转码作业](#)接口。