阿里云
媒体处理

SDK参考

文档版本：20190129

法律声明

阿里云提醒您在阅读或使用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
	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。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
*## *	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 创建AccessKey

请您按照以下步骤创建AccessKey。

1. 登录阿里云官网。
2. 访问 *AccessKey* 管理中心控制台。
3. 创建和管理您的AccessKey。

2 上传SDK

2.1 使用说明

上传SDK提供的功能包括文件列表管理和上传控制。文件列表管理包括文件的增加、删除、取消、恢复、遍历、清空。上传控制包括开始、停止、暂停、恢复。SDK也提供了回调事件，用来监听上传过程中的状态和进度变化。

上传流程

初始化 > 用户选择文件 > 添加文件到列表 > 开始上传 > 上传完成事件。

- 初始化

  初始化时，有2种提供授权的方式：AK和安全令牌。考虑到AK在端上保存的安全性，建议AK方式只用于测试，生产环境使用安全令牌方式。详情参考 开发人员指南 >上传视频文件简介。
- 用户选择文件

  用户选择本地要上传的文件。
- 添加文件到列表

  把用户需要上传的所有文件通过addFile接口添加到列表中。
- 开始上传

  调用start接口就会开始真正的上传进程。
- 上传完成事件

  包含成功和失败两种事件OnUploadSucceed，OnUploadFailed。

概念和说明

- 分片上传和状态

  SDK内部采用的是分片上传机制，状态只在一次执行内有效，如果由于各种原因导致应用退出（例如：关机、关闭浏览器页面、关闭APP、APP异常退出等），需要重新上传。
- 授权过期

  安全令牌是有时效性的，当安全令牌过期后，上传过程会中断，并且没法自主恢复。需要从后端重新获取新的安全令牌，并调用resumeUploadWithAuth函数才能恢复上传。
- 移动端3G/4G<->Wifi切换

为了避免浪费3G/4G网络下的流量，应用判断切换到3G/4G网络时（需要应用自己实现判断），可以调用pause暂停上传。在切换回Wifi网络时（需要应用自己实现判断），调用resume恢复上传。

- 现提供了3种终端的SDK：
  - *HTML5*：可以集成到PC的浏览器中，开发语言JavaScript。
  - *iOS*：可以集成到iOS系统的APP中，开发语言Object-C。
  - *Android*：可以集成到Android系统的APP中，开发语言Java。

功能描述

文件列表管理

接口名称	描述
addFile	添加文件到列表中，文件是按照添加的顺序依次上传
deleteFile	从列表中删除文件
cancelFile	取消列表中的单个文件，但是不会从上传列表中删除，效果就是会跳过这个文件的上传(JavaScript版本不支持)
resumeFile	不是恢复上传,只是恢复之前列表中被取消单个文件的状态(JavaScript版本不支持)
listFiles	获取列表
clearFiles	清除列表，即使是上传中的文件，也会停止上传并清除(JavaScript版本不支持)

上传控制

接口名称	描述
start	开始上传
stop	停止上传
pause	暂停上传(JavaScript版本不支持)
resume	恢复上传(JavaScript版本不支持)
resumeUploadWithToken	安全令牌超时后，使用新的安全令牌来恢复上传

回调事件

事件名称	描述
OnUploadStarted	每个文件开始上传时都会触发。
OnUploadSucceed	上传成功。
OnUploadFailed	上传失败。可恢复型的错误会自动断点续传，例如：网络异常、超时等。不可恢复类型的错误会导致失败，例如：上传凭证错误、文件不存在等
OnUploadProgress	上传进度汇报。在分片上传成功时触发
OnUploadTokenExpired	安全令牌超时。需要从服务重新获取新的安全令牌，并调用resumeUploadWithToken函数恢复上传
OnUploadRetry	上传过程中，状态由正常切换为异常时触发。例如：网络异常，超时等(JavaScript版本不支持)
OnUploadRetryResume	上传过程中，状态由异常中恢复时触发。(JavaScript版本不支持)

2.2 JavaScript版本

安装

上传SDK下载

在页面上引入下面两个JS脚本

```
<script src="aliyun-sdk.min.js"></script>
<script src="vod-sdk-upload-1.0.6.min.js"></script>
```

创建对象初始化：创建VODUpload实例

在这里您需要设置回调函数。

```
var uploader = new VODUpload({
    // 开始上传
    'onUploadstarted': function (uploadInfo) {
      log("onUploadStarted:" + uploadInfo.file.name + ", endpoint:" +
uploadInfo.endpoint + ", bucket:" + uploadInfo.bucket + ", object:" +
uploadInfo.object);
    }
    // 文件上传成功
    'onUploadSucceed': function (uploadInfo) {
      log("onUploadSucceed: " + uploadInfo.file.name + ", endpoint:"
 + uploadInfo.endpoint + ", bucket:" + uploadInfo.bucket + ", object:"
 + uploadInfo.object);
    },
    // 文件上传失败
    'onUploadFailed': function (uploadInfo, code, message) {
      log("onUploadFailed: file:" + uploadInfo.file.name + ",code:" +
code + ", message:" + message);
    },
```

```
    // 文件上传进度，单位：字节
    'onUploadProgress': function (uploadInfo, totalSize, uploadedSize)
 {
        log("onUploadProgress:file:" + uploadInfo.file.name + ",
fileSize:" + totalSize + ", percent:" + Math.ceil(uploadedSize * 100
 / totalSize) + "%");
    },
    // 安全令牌超时
    'onUploadTokenExpired': function () {
        console.log("onUploadTokenExpired");
        // uploader.resumeUploadWithToken(accessKeyId, accessKeySecret
, secretToken, expireTime);
    }
});
```

列表管理

- 添加上传文件

> **说明:**
> 支持的文件大小≤10G。

需要使用标准的input方式让用户选择文件。

```
<form action="">
<input type="file" name="file" id="files" multiple/>
</form>
userData = '';
document.getElementById("files")
     .addEventListener('change', function (event) {
       for(var i=0; i<event.target.files.length; i++) {
         // 逻辑代码
       }
     });
```

获取到用户选择的文件后，添加到上传列表中。

```
uploader.addFile(event.target.files[i], endpoint, bucket, object,
userData);
```

> **说明:**
>
> - event.target.files[i]：用户选择的文件列表,
> - endpoint：OSS的endpoint,
> - bucket：OSS的bucket,
> - object：OSS的object,
> - userData：addFile函数最后的参数userData是一个json对象。

上传时，如何指定媒体的属性（标题、标签、描述、类目、用户自定义数据）？addFile函数最后的参数userData是一个json对象。第一级的Vod是为必须，Vod下有5个属性，示例如下：

```
var userData = '{"Vod":{"Title":"我是标题",
                   "Description":"我是描述",
                   "CateId":"1",
                   "Tags":"tag1,tag2,标签3",
                   "UserData":"user data"}}';
```

- 删除上传文件，`index`对应listFiles接口返回列表中元素的索引。

```
uploader.deleteFile(index);
```

- 取消单个文件上传。

```
uploader.cancelFile(index);
```

- 恢复单个文件上传。

```
uploader.resumeFile(index);
```

- 获取上传文件列表。

```
uploader.listFiles();
 var list = uploader.listFiles();
    for (var i=0; i<list.length; i++) {
        log("file:" + list[i].file.name + ", status:" + list[i].
state + ", endpoint:" + list[i].endpoint + ", bucket:" + list[i].
bucket + ", object:" + list[i].object);
```

```
    }
```

- 清理上传文件列表。

```
uploader.cleanList();
```

上传控制

- 开始上传

```
uploader.startUpload();
```

- 停止上传

```
uploader.stopUpload();
```

- 上传凭证失效后恢复上传

```
uploader.resumeUploadWithToken(accessKeyId, accessKeySecret,
secretToken, expireTime);
```

2.3 Android版本

本节主要介绍Android版本的环境要求，安装、创建VODUpload实例、初始化、列表管理和上传控制。

环境要求

Android系统版本：2.3 及以上

安装

OSS Android SDK

*上传SDK*下载

直接引入jar包

当您下载了VODUpload Android SDK的zip包后，进行以下步骤（对Android studio或者Eclipse都适用）。

1. 解压后在libs目录下得到jar包，目前包括aliyun-oss-sdk-android-xxx.jar、okhttp-2.7.0.jar、okio-2.6.0.jar、aliyun-vod-upload-android-sdk-xxx.jar。
2. 将以上4个jar包导入工程的libs目录。
3. 设置权限。

以下是VODUpload Android SDK所需要的Android权限，请确保您的AndroidManifest.xml文件中已经配置了这些权限，否则，SDK将无法正常工作。

```
<uses-permission android:name="android.permission.INTERNET"></uses-
permission>
<uses-permission android:name="android.permission.ACCESS_NET
WORK_STATE"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE
"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTE
RNAL_STORAGE"></uses-permission>
```

创建VODUpload实例

您需要在这里设置回调函数：

```
VODUploadCallback callback = new VODUploadCallback() {
    /**
    * 文件开始上传时触发
    */
    void onUploadStarted() {;}
    /**
    * 上传成功回调
    */
    void onUploadSucceed(UploadFileInfo info) {;}
    /**
    * 上传失败
    */
    void onUploadFailed(UploadFileInfo info, String code, String
message) {;}
    /**
    * 回调上传进度
    * @param uploadedSize 已上传字节数
    * @param totalSize 总共需要上传字节数
    */
    void onUploadProgress(UploadFileInfo info, long uploadedSize, long
 totalSize) {;}
    /**
    * 上传凭证过期后，会回调这个接口
    * 可在这个回调中获取新的上传，然后调用resumeUploadWithAuth继续上传
    */
    void onUploadTokenExpired() {;}
    /**
    * 上传过程中，状态由正常切换为异常时触发
    */
    void onUploadRetry(String code, String message) {;}
    /**
    * 上传过程中，从异常中恢复时触发
    */
    void onUploadRetryResume() {;}
};
VODUploadClient uploader = new VODUploadClientImpl(getContext());
```

初始化

填写授权信息，有2种方式：

- AK方式

简单但是不够安全，建议您在测试环境下使用。

```
uploader.init("<accessKeyId>", "<accessKeySecret>", callback);
```

- 安全令牌方式

  安全但是较为复杂，建议您在生产环境下使用。安全令牌是临时、有时效性的，所以传递安全令牌是安全的。

```
uploader.init("<accessKeyId>", "<accessKeySecret>", "<secretToken>",
 "<expireTime>", callback);
```

列表管理

- 添加上传文件。

  

  说明:

  支持的文件大小≤4G。

```
uploader.addFile("<uploadFilePath>",
                 "<endpoint>", // 例如杭州区域"http://oss-cn-hangzhou.
aliyuncs.com"
                 "<bucketName>", // 按实际bucket名称填写
                 "<objectKey>");
```

上传时，如何指定媒体的属性（标题、标签、描述、类目、封面URL、用户自定义数据）呢？addFile有一个重载函数，函数最后的参数是一个VodInfo对象。定义如下：

```
private String title;
private String desc;
private Integer cateId;
private List<String> tags;
private String userData;
```

```
private String coverUrl;
```

- 删除上传文件，index对应listFiles接口返回列表中元素的索引。

```
uploader.deleteFile(index);
```

- 取消列表中的单个文件上传。

```
uploader.cancelFile(index);
```

- 恢复列表中的单个文件上传。

```
uploader.resumeFile(index);
```

- 获取上传文件列表。

```
List list = uploader.listFiles();
```

- 清除上传文件列表。

```
upload.clearFiles();
```

上传控制

- 开始上传。

```
uploader.start();
```

- 停止上传。

```
uploader.stop();
```

- 暂停上传。

```
uploader.pause();
```

- 恢复上传。

```
uploader.resume();
```

- 安全令牌失效后恢复上传。

```
uploader.resumeWithToken("<accessKeyId>", "<accessKeySecret>", "<
secretToken>", "<expireTime>");
```

2.4 iOS版本

环境要求

iOS系统版本：iOS 7.0以上。

安装

OSS iOS SDK

上传SDK下载

- 直接引入Framework。

  需要引入OSS iOS SDK framework和VODUpload iOS SDK framework。

  在Xcode中，直接把framework拖入您对应的Target下即可，在弹出框勾选Copy items if needed。

- 工程中引入头文件。

```
#import <VODUpload/VODUploadClient.h>
```

> 说明:
> 引入Framework后，需要在工程Build Settings的Other Linker Flags中加入-ObjC。如果工程此前已经设置过-force_load选项，那么，需要加入-force_load <framework path>/AliyunOSSiOS。

- 兼容IPv6-Only网络。

  OSS移动端SDK为了解决无线网络下域名解析容易遭到劫持的问题，已经引入了HTTPDNS进行域名解析，直接使用IP请求OSS服务端。在IPv6-Only的网络下，可能会遇到兼容性问题。而APP官方近期发布了关于IPv6-only网络环境兼容的APP审核要求，为此，SDK从2.5.0版本开始已经做了兼容性处理。在新版本中，除了-ObjC的设置，还需要引入两个系统库。

```
libresolv.tbd
SystemConfiguration.framework
```

创建VODUpload实例

在这里需要设置回调函数

```
OnUploadStartedListener testUploadStartedCallbackFunc = ^(UploadFile
Info* fileInfo) {
      NSLog(@"upload started .");
  };
 OnUploadSucceedListener testSuccessCallbackFunc = ^(NSString*
filePath){
    NSLog(@"file:%@ upload success!", filePath);
 };
 OnUploadFailedListener testFailedCallbackFunc = ^(NSString* filePath
, NSString* code, NSString* message){
    NSLog(@"failed code = %@, error message = %@", code, message);
 };
 // 单位：字节
 OnUploadProgressListener testProgressCallbackFunc = ^(NSString*
filePath, long uploadedSize, long totalSize) {
```

```
    NSLog(@"progress uploadedSize : %li, totalSize : %li", uploadedSi
ze, totalSize);
 };
 OnUploadTokenExpiredListener testTokenExpiredCallbackFunc = ^{
    NSLog(@"*token expired.");
    // get token and call resmeUploadWithAuth.
 };
 OnUploadRertyListener testUploadRertyListener = ^{
    NSLog(@"retry begin.");
 };
 OnUploadRertyResumeListener testUploadRertyResumeListener = ^{
    NSLog(@"retry resume.");
 };
 VODUploadListener *listener;
 listener = [[VODUploadListener alloc] init];
 listener.started = testUploadStartedCallbackFunc;
 listener.success = testSuccessCallbackFunc;
 listener.failure = testFailedCallbackFunc;
 listener.progress = testProgressCallbackFunc;
 listener.expire = testTokenExpiredCallbackFunc;
 listener.retry = testUploadRertyListener;
 listener.retryResume = testUploadRertyResumeListener;
```

初始化

填写授权信息，有2种方式

- **AK方式**

  简单但是不够安全，建议您在测试环境下使用。

  ```
  VODUploadClient *uploader;
   [uploader init:<accessKeyId>
            accessKeySecret:<accessKeySecret>
            listener:listener];
  ```

- **安全令牌方式**

  安全但是较为复杂，建议您在生产环境下使用。安全令牌是临时、有时效性的，所以传递安全令牌是安全的。

  ```
  VODUploadClient *uploader;
   [uploader init:<accessKeyId>
            accessKeySecret:<accessKeySecret>
            secretToken:<secretToken>
            expireTime:<expireTime>
            listener:listener];
  ```

- **列表管理**

  - **添加上传文件。**

    

    说明:

    支持的文件大小≤4G。

    ```
    [uploader addFile:<uploadFilePath>
    ```

```
        endpoint:<endpoint> //例如: 'http://oss-cn-hangzhou.
aliyuncs.com'
        bucket:<bucketName> //按实际bucket名称填写
        object:<objectKey>];
```

上传时，如何指定媒体的属性（标题、标签、描述、类目、封面URL、用户自定义数据）呢？addFile有一个重载函数，函数最后的参数是一个VodInfo对象。定义如下：

```
@interface VodInfo : NSObject
@property (nonatomic, strong) NSString* title;
@property (nonatomic, strong) NSString* tags;
@property (nonatomic, strong) NSString* desc;
@property (nonatomic, strong) NSNumber* cateId;
@property (nonatomic, strong) NSString* userData;
```

```
@property (nonatomic, strong) NSString* coverUrl;
```

- 删除上传文件。

```
[uploader deleteFile:<index>];
```

- 取消列表中的单个文件上传。

```
[uploader cancelFile:<index>];
```

- 恢复列表中的单个文件上传。

```
[uploader resumeFile:<index>];
```

- 获取上传文件列表。

```
[uploader listFiles];
```

- 清理上传文件列表。

```
[uploader clearFiles];
```

上传控制

- 开始上传。

```
[uploader start];
```

- 停止上传。

```
[uploader stop];
```

- 暂停上传。

```
[uploader pause];
```

- 恢复上传。

```
[uploader resume];
```

- 安全令牌失效后恢复上传。

```
[uploader resumeWithToken:<accessKeyId>
          accessKeySecret:<accessKeySecret>
          secretToken:<secretToken>
          expireTime:<expireTime>]
```

2.5 上传SDK下载

最新版本

- *JavaScript*版本 *1.0.6*，示例代码
- *iOS*版本 *1.0.7*，示例代码

- *android*版本 *1.0.6*，示例代码

1.0.7

- *iOS*版本 *1.0.7*

更新日志

- iOS支持bitcode

1.0.6

- *JavaScript*版本 *1.0.6*，示例代码
- *iOS*版本 *1.0.6*，示例代码
- *android*版本 *1.0.6*，示例代码

更新日志

- android和iOS的列表管理接口(cancelFile,deleteFile,resumeFile)参数使用索引序号替代文件名，方便和列表(listFiles)对应。
- 兼容点播和传统的OSS两种上传模式
- 示例代码增加点播上传模式

1.0.5

- *JavaScript*版本 *1.0.5*，示例代码
- *iOS*版本 *1.0.5*，示例代码
- *android*版本 *1.0.5*，示例代码

更新日志

- 增加暂停、恢复、网络异常事件
- 上传时可以指定视频媒体的标题、标签、封面URL、类目Id、描述、用户自定义数据
- iOS集成OSS的2.6.0版本，支持苹果ATS标准

0.0.4

- *JavaScript*版本 *0.0.4*，示例代码
- *iOS*版本 *0.0.4*，示例代码
- *Android*版本 *0.0.4*，示例代码

更新日志

- 初始版本

3 媒体转码SDK

3.1 SDK 文档简介

本文档介绍媒体处理的SDK安装、使用、示例代码等。

媒体处理SDK基于阿里云SDK，关于阿里云SDK的更多详细帮助请参考：

- 阿里云*SDK*平台
- 阿里云*API*平台

3.2 多地域支持

媒体处理在很多地域开放了服务，详细的地域列表以及SDK对应的RegionId参见：

- 媒体处理的服务地域
- 地域对应的*RegionId*

3.3 Java SDK

3.3.1 前言

媒体处理的Java SDK基于阿里云Java SDK。本文介绍阿里云Java SDK的基本知识。

- 阿里云*Java SDK*快速入门
- 阿里云*Java SDK*使用手册
- 阿里云*Java SDK GitHub*仓库

了解基本知识后，您可以进行媒体处理Java SDK的安装。详情参见 媒体处理 > *SDK*参考 > *Java SDK* > 安装。

3.3.2 安装

本文介绍阿里云Java SDK推荐的Maven安装方式。具体包含2个步骤，首先需要在pom.xml配置文件中添加阿里云Java SDK的Maven仓库，然后再添加媒体处理的依赖。

1. 添加maven仓库。

```
<repositories>
        <repository>
            <id>sonatype-nexus-staging</id>
            <name>Sonatype Nexus Staging</name>
```

```
            <url>https://oss.sonatype.org/service/local/staging/
deploy/maven2/</url>
            <releases>
                <enabled>true</enabled>
            </releases>
            <snapshots>
                <enabled>true</enabled>
            </snapshots>
        </repository>
    </repositories>
```

2. 添加依赖。

   阿里云Java SDK核心库以及媒体处理Java SDK的详细版本：

   - 阿里云*Java SDK*核心库版本
   - 媒体处理*Java SDK*版本

   以3.5.0版本SDK的核心库和2.5.2版本的媒体处理SDK为例：

```
<dependency>
          <groupId>com.aliyun</groupId>
          <artifactId>aliyun-java-sdk-core</artifactId>
          <version>3.5.0</version>
    </dependency>
    <dependency>
          <groupId>com.aliyun</groupId>
          <artifactId>aliyun-java-sdk-mts</artifactId>
          <version>2.5.2</version>
    </dependency>
```

   另外有一个可选的json库依赖。在媒体处理的API中，很多参数都是json定义，java的json库有很多，可以选择熟悉的库，以1.2.46版本的 *fastjson* 为例：

```
<dependency>
        <groupId>com.alibaba</groupId>
        <artifactId>fastjson</artifactId>
        <version>1.2.46</version>
    </dependency>
```

pom.xml完整示例：

```
<?xml version="1.0" encoding="UTF-8"?>
  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http
://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://
maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>mps-demo-project</groupId>
  <artifactId>mps-demo-project</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <repositories>
      <repository>
          <id>sonatype-nexus-staging</id>
          <name>Sonatype Nexus Staging</name>
          <url>https://oss.sonatype.org/service/local/staging/deploy/
maven2/</url>
          <releases>
```

```
            <enabled>true</enabled>
        </releases>
        <snapshots>
            <enabled>true</enabled>
        </snapshots>
    </repository>
</repositories>
<dependencies>
    <dependency>
        <groupId>com.aliyun</groupId>
        <artifactId>aliyun-java-sdk-core</artifactId>
        <version>3.5.0</version>
    </dependency>
    <dependency>
        <groupId>com.aliyun</groupId>
        <artifactId>aliyun-java-sdk-mts</artifactId>
        <version>2.5.2</version>
    </dependency>
    <dependency>
        <groupId>com.alibaba</groupId>
        <artifactId>fastjson</artifactId>
        <version>1.2.46</version>
    </dependency>
</dependencies>
<build>
    <finalName>${artifactId}-${version}</finalName>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>2.3.2</version>
            <configuration>
                <source>1.6</source>
                <target>1.6</target>
            </configuration>
        </plugin>
    </plugins>
</build>
</project>
```

3.3.3 快速入门

本文介绍Java SDK快速入门流程。

1. **创建AcsClient实例。**

```
DefaultProfile profile = DefaultProfile.getProfile(
             mpsRegionId,      // 地域ID
             accessKeyId,      // RAM账号的AccessKey ID
             accessKeySecret); // RAM账号Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);
```

2. **创建request，并设置参数。**

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. **发起API请求并显示返回值。**

```
response = client.getAcsResponse(request);
System.out.println("PipelineName is:" + response.getPipelineList().
get(0).getName());
```

```
System.out.println("PipelineId is:" + response.getPipelineList().get
(0).getId());
```

完整代码

```
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Quick {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String[] mpsRegionIds = new String[] {
            "cn-hangzhou", "cn-beijing","cn-shenzhen", "cn-shanghai",
            "cn-hongkong", "us-west-1", "ap-southeast-1", "ap-
northeast-1",
            "eu-central-1", "ap-south-1"
    };
    public static void main(String[] args) {
        for (String mpsRegionId : mpsRegionIds) {
            System.out.println("region id is:" + mpsRegionId);
            // 创建DefaultAcsClient实例并初始化
            DefaultProfile profile = DefaultProfile.getProfile(
                    mpsRegionId,      // 地域ID
                    accessKeyId,      // RAM账号的AccessKey ID
                    accessKeySecret); // RAM账号Access Key Secret
            IAcsClient client = new DefaultAcsClient(profile);
            // 创建API请求并设置参数
            SearchPipelineRequest request = new SearchPipelineRequest
();
            // 发起请求并处理应答或异常
            SearchPipelineResponse response;
            try {
                response = client.getAcsResponse(request);
                System.out.println("PipelineName is:" + response.
getPipelineList().get(0).getName());
                System.out.println("PipelineId is:" + response.
getPipelineList().get(0).getId());
            } catch (ServerException e) {
                e.printStackTrace();
            } catch (ClientException e) {
                e.printStackTrace();
            }
        }
    }
}
```

3.3.4 转码

1. **创建AcsClient实例。**

```
DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId,      // 地域ID
            accessKeyId,      // RAM账号的AccessKey ID
            accessKeySecret); // RAM账号Access Key Secret
```

```
IAcsClient client = new DefaultAcsClient(profile);
```

2. 创建request，并设置参数。

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. 转码参数。

- Input

```
JSONObject input = new JSONObject();
input.put("Location", ossLocation);
input.put("Bucket", ossBucket);
input.put("Object", URLEncoder.encode(ossInputObject, "utf-8"));
request.setInput(input.toJSONString());
```

- Output

```
String outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8
");
JSONObject output = new JSONObject();
output.put("OutputObject", outputOSSObject);
```

  - Container

```
JSONObject container = new JSONObject();
container.put("Format", "mp4");
output.put("Container", container.toJSONString());
```

  - Video

```
JSONObject video = new JSONObject();
video.put("Codec", "H.264");
video.put("Bitrate", "1500");
video.put("Width", "1280");
video.put("Fps", "25");
output.put("Video", video.toJSONString());
```

  - Audio

```
JSONObject audio = new JSONObject();
audio.put("Codec", "AAC");
audio.put("Bitrate", "128");
audio.put("Channels", "2");
audio.put("Samplerate", "44100");
output.put("Audio", audio.toJSONString());
```

  - TemplateId

```
output.put("TemplateId", templateId);
```

- PipelineId

```
request.setPipelineId(pipelineId);
```

4. 发起API请求并显示返回值。

```
SubmitJobsResponse response;
```

```
response = client.getAcsResponse(request);
System.out.println("RequestId is:"+response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
 System.out.println("JobId is:" + response.getJobResultList().get(0
).getJob().getJobId());
} else {
 System.out.println("SubmitJobs Failed code:" + response.getJobResu
ltList().get(0).getCode() +
                    " message:" + response.getJobResultList().get(0
).getMessage());
}
```

完整代码

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class SimpleTranscode {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String templateId = "S00000001-200010";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    public static void main(String[] args) {
        // 创建DefaultAcsClient实例并初始化
        DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // 地域ID
                accessKeyId,      // RAM账号的AccessKey ID
                accessKeySecret); // RAM账号Access Key Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建API请求并设置参数
        SubmitJobsRequest request = new SubmitJobsRequest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
            input.put("Object", URLEncoder.encode(ossInputObject, "utf
-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("input URL encode failed");
        }
        request.setInput(input.toJSONString());
        // Output
        String outputOSSObject;
        try {
            outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-
8");
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("output URL encode failed");
        }
        JSONObject output = new JSONObject();
```

```
        output.put("OutputObject", outputOSSObject);
        // Ouput->Container
        JSONObject container = new JSONObject();
        container.put("Format", "mp4");
        output.put("Container", container.toJSONString());
        // Ouput->Video
        JSONObject video = new JSONObject();
        video.put("Codec", "H.264");
        video.put("Bitrate", "1500");
        video.put("Width", "1280");
        video.put("Fps", "25");
        output.put("Video", video.toJSONString());
        // Ouput->Audio
        JSONObject audio = new JSONObject();
        audio.put("Codec", "AAC");
        audio.put("Bitrate", "128");
        audio.put("Channels", "2");
        audio.put("Samplerate", "44100");
        output.put("Audio", audio.toJSONString());
        // Ouput->TemplateId
        output.put("TemplateId", templateId);
        JSONArray outputs = new JSONArray();
        outputs.add(output);
        request.setOutputs(outputs.toJSONString());
        request.setOutputBucket(ossBucket);
        request.setOutputLocation(ossLocation);
        // PipelineId
        request.setPipelineId(pipelineId);
        // 发起请求并处理应答或异常
        SubmitJobsResponse response;
        try {
            response = client.getAcsResponse(request);
            System.out.println("RequestId is:"+response.getRequestId
());
            if (response.getJobResultList().get(0).getSuccess()) {
                System.out.println("JobId is:" + response.getJobResu
ltList().get(0).getJob().getJobId());
            } else {
                System.out.println("SubmitJobs Failed code:" +
response.getJobResultList().get(0).getCode() +
                                   " message:" + response.getJobResu
ltList().get(0).getMessage());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

3.3.5 水印

1. **创建AcsClient实例。**

```
DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
```

```
IAcsClient client = new DefaultAcsClient(profile);
```

2. 创建request，并设置参数。

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. 设置转码参数。

- 图片水印

```
// Image Watermark
   JSONObject imageWatermarkInput = new JSONObject();
   imageWatermarkInput.put("Location", ossLocation);
   imageWatermarkInput.put("Bucket", ossBucket);
   try {
       imageWatermarkInput.put("Object", URLEncoder.encode(
imageWatermarkObject, "utf-8"));
   } catch (UnsupportedEncodingException e) {
       throw new RuntimeException("imageWatetmark Input URL encode
 failed");
   }
   JSONObject imageWatermark = new JSONObject();
   imageWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
   imageWatermark.put("Type", "Image");
   imageWatermark.put("InputFile", imageWatermarkInput);
   imageWatermark.put("ReferPos", "TopRight");
   imageWatermark.put("Width", "0.05");
   imageWatermark.put("Dx", "0");
   imageWatermark.put("Dy", "0");
```

- 文字水印

```
// Text Watermark
   JSONObject textConfig = new JSONObject();
   textConfig.put("Content", "5rWL6K+V5paH5a2X5rC05Y2w");
   textConfig.put("FontName", "SimSun");
   textConfig.put("FontSize", "16");
   textConfig.put("FontColor", "Red");
   textConfig.put("FontAlpha", "0.5");
   textConfig.put("Top", "10");
   textConfig.put("Left", "10");
   JSONObject textWatermark = new JSONObject();
   textWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
   textWatermark.put("Type", "Text");
   textWatermark.put("TextWaterMark", textConfig.toJSONString());
```

- 视频水印

```
// Video Watermark
   JSONObject videoWatermarkInput = new JSONObject();
   videoWatermarkInput.put("Location", ossLocation);
   videoWatermarkInput.put("Bucket", ossBucket);
   try {
       videoWatermarkInput.put("Object", URLEncoder.encode(
videoWatermarkObject, "utf-8"));
   } catch (UnsupportedEncodingException e) {
       throw new RuntimeException("videoWatetmark Input URL encode
 failed");
   }
   JSONObject videoWatermark = new JSONObject();
   videoWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
```

```
    videoWatermark.put("Type", "Image");
    videoWatermark.put("InputFile", videoWatermarkInput);
    videoWatermark.put("ReferPos", "BottomLeft");
    videoWatermark.put("Height", "240");
    videoWatermark.put("Dx", "0");
    videoWatermark.put("Dy", "0");
```

4. 发起API请求并显示返回值。

```
SubmitJobsResponse response;
response = client.getAcsResponse(request);
System.out.println("RequestId is:"+response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
    System.out.println("JobId is:" + response.getJobResultList().get
(0).getJob().getJobId());
} else {
    System.out.println("SubmitJobs Failed code:" + response.
getJobResultList().get(0).getCode() +
                       " message:" + response.getJobResultList().get
(0).getMessage());
}
```

完整代码

```
package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Watermark {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String watermarkTemplateId = "xxx";
    private static String templateId = "S00000001-200030";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "presigned";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    private static String imageWatermarkObject = "logo.png";
    private static String videoWatermarkObject = "logo.mov";
    public static void main(String[] args) {
        // DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // request
        SubmitJobsRequest request = new SubmitJobsRequest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
```

```
            input.put("Object", URLEncoder.encode(ossInputObject, "utf
-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("input URL encode failed");
        }
        request.setInput(input.toJSONString());
        // Output
        String outputOSSObject;
        try {
            outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-
8");
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("output URL encode failed");
        }
        JSONObject output = new JSONObject();
        output.put("OutputObject", outputOSSObject);
        // Ouput->TemplateId
        output.put("TemplateId", templateId);
        // Image Watermark
        JSONObject imageWatermarkInput = new JSONObject();
        imageWatermarkInput.put("Location", ossLocation);
        imageWatermarkInput.put("Bucket", ossBucket);
        try {
            imageWatermarkInput.put("Object", URLEncoder.encode(
imageWatermarkObject, "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("imageWatetmark Input URL
encode failed");
        }
        JSONObject imageWatermark = new JSONObject();
        imageWatermark.put("WaterMarkTemplateId", watermarkTemplateId
);
        imageWatermark.put("Type", "Image");
        imageWatermark.put("InputFile", imageWatermarkInput);
        imageWatermark.put("ReferPos", "TopRight");
        imageWatermark.put("Width", "0.05");
        imageWatermark.put("Dx", "0");
        imageWatermark.put("Dy", "0");
        // Text Watermark
        JSONObject textConfig = new JSONObject();
        textConfig.put("Content", "5rWL6K+V5paH5a2X5rC05Y2w");
        textConfig.put("FontName", "SimSun");
        textConfig.put("FontSize", "16");
        textConfig.put("FontColor", "Red");
        textConfig.put("FontAlpha", "0.5");
        textConfig.put("Top", "10");
        textConfig.put("Left", "10");
        JSONObject textWatermark = new JSONObject();
        textWatermark.put("WaterMarkTemplateId", watermarkTemplateId);
        textWatermark.put("Type", "Text");
        textWatermark.put("TextWaterMark", textConfig.toJSONString());
        // Video Watermark
        JSONObject videoWatermarkInput = new JSONObject();
        videoWatermarkInput.put("Location", ossLocation);
        videoWatermarkInput.put("Bucket", ossBucket);
        try {
            videoWatermarkInput.put("Object", URLEncoder.encode(
videoWatermarkObject, "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("videoWatetmark Input URL
encode failed");
        }
        JSONObject videoWatermark = new JSONObject();
```

```
        videoWatermark.put("WaterMarkTemplateId", watermarkTemplateId
);
        videoWatermark.put("Type", "Image");
        videoWatermark.put("InputFile", videoWatermarkInput);
        videoWatermark.put("ReferPos", "BottomLeft");
        videoWatermark.put("Height", "240");
        videoWatermark.put("Dx", "0");
        videoWatermark.put("Dy", "0");
        // Output->Watermarks
        JSONArray watermarks = new JSONArray();
        watermarks.add(imageWatermark);
        watermarks.add(textWatermark);
        watermarks.add(videoWatermark);
        output.put("WaterMarks", watermarks.toJSONString());
        // Outputs
        JSONArray outputs = new JSONArray();
        outputs.add(output);
        request.setOutputs(outputs.toJSONString());
        request.setOutputBucket(ossBucket);
        request.setOutputLocation(ossLocation);
        // PipelineId
        request.setPipelineId(pipelineId);
        // call api
        SubmitJobsResponse response;
        try {
            response = client.getAcsResponse(request);
            System.out.println("RequestId is:"+response.getRequestId
());
            if (response.getJobResultList().get(0).getSuccess()) {
                System.out.println("JobId is:" + response.getJobResu
ltList().get(0).getJob().getJobId());
            } else {
                System.out.println("SubmitJobs Failed code:" +
response.getJobResultList().get(0).getCode() +
                                   " message:" + response.getJobResu
ltList().get(0).getMessage());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

3.3.6 截图

1. **创建AcsClient实例。**

```
DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);
```

2. **创建request，并设置参数。**

```
SubmitSnapshotJobRequest request = new SubmitSnapshotJobRequest();
```

3. **设置截图参数。**

- Input

```
JSONObject input = new JSONObject();
   input.put("Location", ossLocation);
   input.put("Bucket", ossBucket);
   try {
       input.put("Object", URLEncoder.encode(ossInputObject, "utf-
8"));
   } catch (UnsupportedEncodingException e) {
       throw new RuntimeException("input URL encode failed");
   }
   request.setInput(input.toJSONString());
```

- SnapshotConfig

```
JSONObject snapshotConfig = new JSONObject();
```

  - OutputFile

```
JSONObject output = new JSONObject();
output.put("Location", ossLocation);
output.put("Bucket", ossBucket);
try {
    output.put("Object", URLEncoder.encode(ossOutputObject, "
utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("output URL encode failed");
}
snapshotConfig.put("OutputFile", output.toJSONString());
```

  - Time

```
snapshotConfig.put("Time", "2");
```

  - Interval/Num

```
snapshotConfig.put("Interval", "2");
snapshotConfig.put("Num", "3");
```

  - Width/Height

```
snapshotConfig.put("Height", "360");
```

4. 发起API请求并显示返回值。

```
SubmitSnapshotJobResponse response;
response = client.getAcsResponse(request);
            System.out.println("RequestId is:"+response.getRequestId
());
System.out.println("JobId is:" + response.getSnapshotJob().getId());
System.out.println(String.format(
                                        "http://%s.%s.aliyuncs.com/
output_00001.jpg",
                                        ossBucket,
                                        ossLocation));
System.out.println(String.format(
                                        "http://%s.%s.aliyuncs.com/
output_00002.jpg",
```

```
                                                ossBucket,
                                                ossLocation));
System.out.println(String.format(
                                                "http://%s.%s.aliyuncs.com/
output_00003.jpg",
                                                ossBucket,
                                                ossLocation));
```

完整代码

```
package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Snapshot {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output_{Count}.jpg";
    public static void main(String[] args) {
        // DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // request
        SubmitSnapshotJobRequest request = new SubmitSnapshotJobReq
uest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
            input.put("Object", URLEncoder.encode(ossInputObject, "utf
-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("input URL encode failed");
        }
        request.setInput(input.toJSONString());
        // SnapshotConfig
        JSONObject snapshotConfig = new JSONObject();
        // SnapshotConfig->OutputFile
        JSONObject output = new JSONObject();
        output.put("Location", ossLocation);
        output.put("Bucket", ossBucket);
        try {
            output.put("Object", URLEncoder.encode(ossOutputObject, "
utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("output URL encode failed");
        }
```

```
        snapshotConfig.put("OutputFile", output.toJSONString());
        // SnapshotConfig->Time
        snapshotConfig.put("Time", "2");
        // SnapshotConfig->Interval/Num
        snapshotConfig.put("Interval", "2");
        snapshotConfig.put("Num", "3");
        // SnapshotConfig->Width/Height
        snapshotConfig.put("Height", "360");
        // SnapshotConfig
        request.setSnapshotConfig(snapshotConfig.toJSONString());
        // PipelineId
        request.setPipelineId(pipelineId);
        // call api
        SubmitSnapshotJobResponse response;
        try {
            response = client.getAcsResponse(request);
            System.out.println("RequestId is:"+response.getRequestId
());
            System.out.println("JobId is:" + response.getSnapshotJob
().getId());
            System.out.println(String.format(
                                        "http://%s.%s.aliyuncs.com/
output_00001.jpg",
                                        ossBucket,
                                        ossLocation));
            System.out.println(String.format(
                                        "http://%s.%s.aliyuncs.com/
output_00002.jpg",
                                        ossBucket,
                                        ossLocation));
            System.out.println(String.format(
                                        "http://%s.%s.aliyuncs.com/
output_00003.jpg",
                                        ossBucket,
                                        ossLocation));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

3.3.7 HLS数据加密

使用场景

- **HLS标准数据加密适用于“对视频进行简单的保护”的，可以简单的防止非法下载和非法传播。**

使用限制

- **HLS标准数据加密需要使用SubmitJobs接口。**
- **工作流中不可使用HLS标准数据加密。**

示例代码依赖

- **MPS SDK详情参见 安装。**

- 其他依赖。

```
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.25</version
</dependency>
<dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.9</version>
</dependency>
```

示例代码

```
package com.aliyun
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.mts.model.v20140618.QueryJobListRequest;
import com.aliyuncs.mts.model.v20140618.QueryJobListResponse;
import com.aliyuncs.mts.model.v20140618.SubmitJobsRequest;
import com.aliyuncs.mts.model.v20140618.SubmitJobsResponse;
import com.aliyuncs.profile.DefaultProfile;
import org.apache.commons.codec.binary.Base64;
public class DataEncryptionDemo {
    private DefaultAcsClient client = null;
    private final String REGION = "cn-shanghai"; //按需配置
    private final String MTS_ENDPOINT = "mts.cn-shanghai.aliyuncs.com
"; //按需配置
    private final String ID="idid"; //按需配置
    private final String KEY ="keykey"; //按需配置
    private final String LOCATION = "oss-cn-shanghai"; //按需配置
    private final String INPUT_BUCKET = "input-bucket"; //按需配置
    private final String OUTPUT_BUCKET = "output-bucket"; //按需配置
    private final String PIPELINE_ID = "pipelineId"; //按需配置
    public DataEncryptionDemo() throws ClientException {
        DefaultProfile.addEndpoint(REGION, REGION, "Mts", MTS_ENDPOINT
);
        this.client = new DefaultAcsClient(DefaultProfile.getProfile(
REGION, ID, KEY));
    }
    private JSONObject getInputFile() {
        JSONObject inputFile = new JSONObject();
        inputFile.put("Location", LOCATION);
        inputFile.put("Bucket", INPUT_BUCKET);
        inputFile.put("Object", "uploadvideo/test.flv");
        return inputFile;
    }
    private JSONArray getOutputs() {
        JSONArray outputs = new JSONArray();
        outputs.add(getOutput());
        return outputs;
    }
    private JSONObject getOutput() {
        JSONObject output = new JSONObject();
        output.put("OutputObject", "BaseTest/hls-encryption.m3u8");
        output.put("TemplateId", "S00000001-100020");
        output.put("Encryption", getEncryptionConfigs());
        return output;
    }
```

```
    private JSONObject getEncryptionConfigs() {
        JSONObject encryption = new JSONObject();
        encryption.put("Type", "hls-aes-128");
        encryption.put("Key", Base64.encodeBase64URLSafeString("
encryptionkey123".getBytes()));
        encryption.put("KeyUri", Base64.encodeBase64URLSafeString("
http://demo.aliyuncs.com/document/hls128.key".getBytes()));
        encryption.put("KeyType", "Base64");
        return encryption;
    }
    private String submitJobs() throws ClientException {
        JSONObject inputFile = getInputFile();
        SubmitJobsRequest request = new SubmitJobsRequest();
        request.setInput(inputFile.toJSONString());
        request.setOutputLocation(LOCATION);
        request.setOutputBucket(OUTPUT_BUCKET);
        request.setOutputs(getOutputs().toJSONString());
        request.setPipelineId(PIPELINE_ID);
        SubmitJobsResponse reponse = this.client.getAcsResponse(
request);
        System.out.println(JSONObject.toJSONString(reponse.getJobResu
ltList()));
        return reponse.getJobResultList().get(0).getJob().getJobId();
    }
    public static void main(String[] args) throws ClientException {
        DataEncryptionDemo demo = new DataEncryptionDemo();
        String jobId= demo.submitJobs();
    }
 }
```

3.3.8 创建HLS标准加密工作流

简介

示例调用API进行创建HLS标准加密工作流。创建工作流，参见 新增媒体工作流。

示例代码依赖

- MPS SDK，参见 安装。
- 其他依赖。

```
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.25</version
</dependency>
```

示例代码

```
package com.aliyun.smallcode;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.mts.model.v20140618.AddMediaWorkflowRequest;
import com.aliyuncs.mts.model.v20140618.AddMediaWorkflowResponse;
import com.aliyuncs.profile.DefaultProfile;
public class MediaHls {
    final String REGION_ID = "<region>";
```

```
    final String ACCESS_KEY_ID = "<accessKeyId>";
    final String ACCESS_KEY_SECRET = "<accessKeySecret>";
    final String PIPELINE_ID = "<PipelineId>";
    final String TEMPLATE_ID = "S00000001-100020"; //转码模版ID，m3u8模
版，按需配置
    final String OSS_LOCATION = "<OssLocation>";
    final String INPUT_BUCKET = "<InputBucket>"; //输入bucket
    final String INPUT_PATH = "<InputPath>"; //如 "HLS-Encryption"
    final String OUTPUT_BUCKET = "<OutputBucket>"; //输出bucket
    final String ENCRYPTION_TYPE = "hls-aes-128";
    final String HLS_KEY_URI = "<解密密钥的URI>"; //如http://decrypt.
testdomain.com
    final String ACT_START = "Act-Start";
    final String ACT_ENCRYPTION = "Act-HLS-Encryption";
    final String ACT_REPORT = "Act-Report";
    private DefaultAcsClient client;
    public MediaHls() {
        DefaultProfile profile = DefaultProfile.getProfile(REGION_ID,
ACCESS_KEY_ID, ACCESS_KEY_SECRET);
        this.client = new DefaultAcsClient(profile);
    }
    public AddMediaWorkflowResponse addMediaWorkflow() throws
ClientException {
        AddMediaWorkflowRequest request = new AddMediaWorkflowRequest
();
        request.setTopology(createWorkflow().toJSONString());
        request.setName("HLS加密工作流");
        return this.client.getAcsResponse(request);
    }
    private JSONObject createWorkflow() {
        JSONObject workflow = new JSONObject();
        JSONObject activities = new JSONObject();
        activities.put(ACT_START, createStartActivity());
        activities.put(ACT_ENCRYPTION, createTrancodeActivity());
        activities.put(ACT_REPORT, createReportActivity());
        workflow.put("Activities", activities);
        workflow.put("Dependencies", createDependencies());
        return workflow;
    }
    private JSONObject createStartActivity() {
        JSONObject startActivity = new JSONObject();
        startActivity.put("Name", ACT_START);
        startActivity.put("Type", "Start");
        startActivity.put("Parameters", buildStartParameters());
        return startActivity;
    }
    private JSONObject buildStartParameters() {
        JSONObject parameters = new JSONObject();
        parameters.put("PipelineId", PIPELINE_ID);
        parameters.put("InputFile", buildInputFile());
        return parameters;
    }
    private JSONObject buildInputFile() {
        JSONObject inputFile = new JSONObject();
        inputFile.put("Bucket", INPUT_BUCKET);
        inputFile.put("Location", OSS_LOCATION);
        inputFile.put("ObjectPrefix", INPUT_PATH);
        return inputFile;
    }
    private JSONObject createTrancodeActivity() {
        JSONObject transcodeActivity = new JSONObject();
        transcodeActivity.put("Name", ACT_ENCRYPTION);
        transcodeActivity.put("Type", "Transcode");
```

```
        transcodeActivity.put("Parameters", buildTranscodeParameters
());
        return transcodeActivity;
    }
    private JSONObject buildTranscodeParameters() {
        JSONObject transcodeParamters = new JSONObject();
        transcodeParamters.put("OutputBucket", OUTPUT_BUCKET);
        transcodeParamters.put("OutputLocation", OSS_LOCATION);
        transcodeParamters.put("Outputs", buildOutputsConfig());
        return transcodeParamters;
    }
    private JSONArray buildOutputsConfig() {
        JSONArray outputs = new JSONArray();
        JSONObject output = new JSONObject();
        output.put("ObjectRegex", ACT_ENCRYPTION + "/{RunId}/{FileName
}");
        output.put("TemplateId", TEMPLATE_ID);
        output.put("Encryption", buildEncryption());
        outputs.add(output);
        return outputs;
    }
    private JSONObject buildEncryption() {
        JSONObject encryption = new JSONObject();
        encryption.put("Type", ENCRYPTION_TYPE);
        encryption.put("KeyUri", HLS_KEY_URI);
        return encryption;
    }
    private JSONObject createReportActivity() {
        JSONObject reportActivity = new JSONObject();
        reportActivity.put("Name", ACT_REPORT);
        reportActivity.put("Parameters", new JSONObject());
        reportActivity.put("Type", "Report");
        return  reportActivity;
    }
    private JSONObject createDependencies() {
        JSONObject dependencies = new JSONObject();
        JSONArray subActivityOfStart = new JSONArray();
        subActivityOfStart.add(ACT_ENCRYPTION);
        dependencies.put(ACT_START, subActivityOfStart);
        JSONArray subActivityOfTranscode = new JSONArray();
        subActivityOfTranscode.add(ACT_REPORT);
        dependencies.put(ACT_ENCRYPTION, subActivityOfTranscode);
        dependencies.put(ACT_REPORT, new JSONArray());
        return dependencies;
    }
    public static void main(String[] args) throws ClientException {
        MediaHls mediaHls = new MediaHls();
        AddMediaWorkflowResponse response = mediaHls.addMediaWorkflow
();
        System.out.println(JSONObject.toJSONString(response));
    }
```

```
}
```

3.3.9 管道管理

在开通服务时，系统会自动创建一个用户管道。您还可以通过一系列接口管理管道（pipeline）。

例如：SearchPipeline、QueryPipelineList、UpdatePipeline。

搜索管道

可以直接通过SearchPipeline接口搜索管道信息。

```
String region = "<region>";
    String accessKeyId = "<accessKeyId>";
    String accessKeySecret = "<accessKeySecret>";
    DefaultProfile profile = DefaultProfile.getProfile(region,
accessKeyId, accessKeySecret);
    DefaultAcsClient client = new DefaultAcsClient(profile);
    SearchPipelineRequest request = new SearchPipelineRequest();
    // 如果出错，可能会抛出ClientException或ServerException异常
    SearchPipelineResponse response = client.getAcsResponse(request);
    List<SearchPipelineResponse.Pipeline> pipelines = response.
getPipelineList();
    for (SearchPipelineResponse.Pipeline pipeline : pipelines) {
        System.out.println("pipeline id:" + pipeline.getId() + ", name
:" + pipeline.getName() + ", state:" + pipeline.getState());
    }
```

查询管道

如果已经知道pipelineId，可以通过pipelineId调用QueryPipelineList接口查询管道信息。

```
String region = "<region>";
    String accessKeyId = "<accessKeyId>";
    String accessKeySecret = "<accessKeySecret>";
    // 已知的管道ID，多个管道用','分隔
    String pipelineIds = "<pipelineIds>";
    DefaultProfile profile = DefaultProfile.getProfile(region,
accessKeyId, accessKeySecret);
    DefaultAcsClient client = new DefaultAcsClient(profile);
    QueryPipelineListRequest request = new QueryPipelineListRequest();
    request.setPipelineIds(pipelineId);
    // 如果出错，可能会抛出ClientException或ServerException异常
    QueryPipelineListResponse response = client.getAcsResponse(request
);
    List<QueryPipelineListResponse.Pipeline> pipelines = response.
getPipelineList();
    for (QueryPipelineListResponse.Pipeline pipeline : pipelines) {
        System.out.println("pipeline id:" + pipeline.getId() + ", name
:" + pipeline.getName() + ", state:" + pipeline.getState());
    }
```

更新管道

通过UpdatePipeline更新管道信息，包括更新管道名称，状态。管道的状态包括Active、Paused。

```
String region = "<region>";
```

```
    String accessKeyId = "<accessKeyId>";
    String accessKeySecret = "<accessKeySecret>";
    DefaultProfile profile = DefaultProfile.getProfile(region,
accessKeyId, accessKeySecret);
    DefaultAcsClient client = new DefaultAcsClient(profile);
    // 检索用户管道，默认只有一个。
    SearchPipelineRequest searchPipelineRequest = new SearchPipe
lineRequest();
    // 如果出错，可能会抛出ClientException或ServerException异常
    SearchPipelineResponse searchPipelineResponse = client.getAcsResp
onse(searchPipelineRequest);
    List<SearchPipelineResponse.Pipeline> pipelines = searchPipe
lineResponse.getPipelineList();
    SearchPipelineResponse.Pipeline queryPipeline = pipelines.get(0);
    // 更新管道状态
    UpdatePipelineRequest request = new UpdatePipelineRequest();
    request.setPipelineId(queryPipeline.getId());
    request.setState("Paused");
    request.setName(queryPipeline.getName());
    // 如果出错，会抛ClientException异常
    UpdatePipelineResponse response = client.getAcsResponse(request);
    UpdatePipelineResponse.Pipeline pipeline = response.getPipeline();
    System.out.println("pipeline id:" + pipeline.getId() + ", name:"
 + pipeline.getName() + ", state:" + pipeline.getState());
```

3.3.10 查询媒体-使用OSS文件地址

如果您在不知道媒体ID（直播走工作流转点播），可以通过媒体的输入地址进行媒体信息的查询，接口为 QueryMediaListByURL。

```
package com.aliyun.mts.api.demo
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.mts.model.v20140618.QueryMediaListByURLRequest;
import com.aliyuncs.mts.model.v20140618.QueryMediaListByURLResponse;
import com.aliyuncs.profile.DefaultProfile;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
public class QueryMediaListByURLDemo {
    private DefaultAcsClient client = null;
    private final String REGION = "<region>";
    private final String ID="<accessKeyId>";
    private final String KEY ="<accessKeySecret>";
    public QueryMediaListByURLDemo() throws ClientException {
        this.client = new DefaultAcsClient(DefaultProfile.getProfile(
REGION, ID, KEY));
    }
    //根据视频源OSS地址查询媒体信息， 如：媒体ID，媒体状态及其他属性
    private void queryMediaListByURL() throws ClientException,
UnsupportedEncodingException {
        QueryMediaListByURLRequest request = new QueryMediaListByURLR
equest();
        String ossHost = 'http://<input-bucket>.<region>.aliyuncs.com
/';
        String ossObject = "test/笑傲江湖.mp4";
        //ossObject需要符合rfc3986标准
        String rfc3986Object = encodeByRFC3986(ossObject);
        request.setFileURLs(ossHost + rfc3986Object);
        QueryMediaListByURLResponse response = this.client.getAcsResp
onse(request);
```

```
        System.out.println(JSONObject.toJSONString(response.getMediaLi
st()));
    }
    private String encodeByRFC3986(String object) throws Unsupporte
dEncodingException {
        StringBuilder builder = new StringBuilder();
        String[] segments = object.split("/");
        for (int i = 0; i < segments.length; i++) {
            builder.append(percentEncode(segments[i]));
            if (i != segments.length - 1) {
                builder.append("/");
            }
        }
        return builder.toString();
    }
    private static String percentEncode(String value) throws
UnsupportedEncodingException {
        if (value == null)
            return null;
        return URLEncoder.encode(value, "UTF-8").replace("+", "%20").
replace("*", "%2A").replace("%7E", "~");
    }
    public static void main(String[] args) {
        try {
            QueryMediaListByURLDemo demo = new QueryMediaListByURLDemo
();
            demo.queryMediaListByURL();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

3.3.11 新增媒体

新增视频文件到媒体库，可以指定工作流ID触发工作流处理视频文件：

```
package com.aliyun.mts;
    import com.alibaba.fastjson.JSONObject;
    import com.aliyuncs.DefaultAcsClient;
    import com.aliyuncs.exceptions.ClientException;
    import com.aliyuncs.exceptions.ServerException;
    import com.aliyuncs.mts.model.v20140618.AddMediaRequest;
    import com.aliyuncs.mts.model.v20140618.AddMediaResponse;
    import com.aliyuncs.profile.DefaultProfile;
    import org.apache.commons.lang.exception.ExceptionUtils;
    public class AddMedia {
        //Step 1 .set region: cn-hangzhou、cn-shenzhen、cn-shanghai、cn
-beijing
        private static final String REGION = "cn-shenzhen";
        private static final String OSS_REGION = "oss-cn-shenzhen";
        private static final String mtsEndpoint = "mts." + REGION + ".
aliyuncs.com";
        //Step 2.set accesskey & keySecret
        private static String accessKeyId = "";
        private static String accessKeySecret = "";
        private static DefaultAcsClient aliyunClient;
        static {
            try {
                DefaultProfile.addEndpoint(REGION, REGION, "Mts",
mtsEndpoint);
            } catch (ClientException e) {
```

```
                System.out.print(ExceptionUtils.getStackTrace(e));
                System.exit(1);
            }
            aliyunClient = new DefaultAcsClient(DefaultProfile.
getProfile(REGION, accessKeyId, accessKeySecret));
        }
        public static void main(String[] args) throws ClientException
 {
            AddMediaRequest request = new AddMediaRequest();
            request.setFileURL("http://mtb-sz-in.oss-cn-shenzhen.
aliyuncs.com/media/r180-ABC.mp4");
            request.setMediaWorkflowId("829bed0300994057a49e4f16de957e
34");
            try {
                AddMediaResponse response = aliyunClient.getAcsResp
onse(request);
                System.out.println(JSONObject.toJSONString(response));
            } catch (ServerException e) {
                System.out.println("Code:" + e.getErrCode() + " Msg:"
 + e.getMessage());
            } catch (ClientException e) {
                System.out.println("Code:" + e.getErrCode() + " Msg:"
 + e.getMessage());
            }
        }
    }
```

3.3.12 视频拼接和简单剪辑

1. **创建AcsClient实例。**

```
DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);
```

2. **创建request，并设置参数。**

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. **设置转码参数。**

   - **Input**

```
JSONObject input = new JSONObject();
   input.put("Location", ossLocation);
   input.put("Bucket", ossBucket);
   try {
       input.put("Object", URLEncoder.encode(headObject, "utf-8
"));
   } catch (UnsupportedEncodingException e) {
       throw new RuntimeException("input URL encode failed");
   }
   request.setInput(input.toJSONString());
```

   - **Output**

```
String outputOSSObject;
   try {
```

```
        outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8
");
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("output URL encode failed");
    }
    JSONObject output = new JSONObject();
    output.put("OutputObject", outputOSSObject);
    // Ouput->TemplateId
    output.put("TemplateId", templateId);
```

- **Video**

```
JSONObject video = new JSONObject();
 video.put("Width", "1280");
 video.put("Height", "720");
 output.put("Video", video.toJSONString());
```

- **MergeList**

```
JSONObject mergeVideo = new JSONObject();
String mergeVideoURL;
try {
    mergeVideoURL = String.format(
                                "http://%s.%s.aliyuncs.com/%s",
                                ossBucket,
                                ossLocation,
                                URLEncoder.encode(ossInputOb
ject, "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeVideoURL encode failed");
}
mergeVideo.put("MergeURL", mergeVideoURL);
JSONObject mergeTail = new JSONObject();
String mergeTailURL;
try {
    mergeTailURL = String.format(
                                "http://%s.%s.aliyuncs.com/%s",
                                ossBucket,
                                ossLocation,
                                URLEncoder.encode(tailObject, "
utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeTailURL encode failed");
}
mergeTail.put("MergeURL", mergeTailURL);
JSONArray mergeList = new JSONArray();
mergeList.add(mergeVideo);
mergeList.add(mergeTail);
output.put("MergeList", mergeList.toJSONString());
```

4. 发起API请求并显示返回值。

```
SubmitJobsResponse response;
response = client.getAcsResponse(request);
System.out.println("RequestId is:"+response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
    System.out.println("JobId is:" + response.getJobResultList().get
(0).getJob().getJobId());
} else {
    System.out.println("SubmitJobs Failed code:" + response.
getJobResultList().get(0).getCode() +
```

```
                            " message:" + response.getJobResultList().get
(0).getMessage());
}
```

完整代码

```
package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class Merge {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String templateId = "S00000001-200030";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    private static String headObject = "head.mp4";
    private static String tailObject = "tail.mp4";
    public static void main(String[] args) {
        // DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // request
        SubmitJobsRequest request = new SubmitJobsRequest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
            input.put("Object", URLEncoder.encode(headObject, "utf-8
"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("input URL encode failed");
        }
        request.setInput(input.toJSONString());
        // Output
        String outputOSSObject;
        try {
            outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-
8");
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("output URL encode failed");
        }
        JSONObject output = new JSONObject();
        output.put("OutputObject", outputOSSObject);
        // Ouput->TemplateId
        output.put("TemplateId", templateId);
        // Ouput->Video
        JSONObject video = new JSONObject();
```

```
        video.put("Width", "1280");
        video.put("Height", "720");
        output.put("Video", video.toJSONString());
        // Output->MergeList
        JSONObject mergeVideo = new JSONObject();
        String mergeVideoURL;
        try {
            mergeVideoURL = String.format(
                                        "http://%s.%s.aliyuncs.com/%s
",
                                        ossBucket,
                                        ossLocation,
                                        URLEncoder.encode(ossInputOb
ject, "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("mergeVideoURL encode failed");
        }
        mergeVideo.put("MergeURL", mergeVideoURL);
        JSONObject mergeTail = new JSONObject();
        String mergeTailURL;
        try {
            mergeTailURL = String.format(
                                        "http://%s.%s.aliyuncs.com/%s
",
                                        ossBucket,
                                        ossLocation,
                                        URLEncoder.encode(tailObject,
 "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("mergeTailURL encode failed");
        }
        mergeTail.put("MergeURL", mergeTailURL);
        JSONArray mergeList = new JSONArray();
        mergeList.add(mergeVideo);
        mergeList.add(mergeTail);
        output.put("MergeList", mergeList.toJSONString());
        // Outputs
        JSONArray outputs = new JSONArray();
        outputs.add(output);
        request.setOutputs(outputs.toJSONString());
        request.setOutputBucket(ossBucket);
        request.setOutputLocation(ossLocation);
        // PipelineId
        request.setPipelineId(pipelineId);
        // call api
        SubmitJobsResponse response;
        try {
            response = client.getAcsResponse(request);
            System.out.println("RequestId is:"+response.getRequestId
());
            if (response.getJobResultList().get(0).getSuccess()) {
                System.out.println("JobId is:" + response.getJobResu
ltList().get(0).getJob().getJobId());
            } else {
                System.out.println("SubmitJobs Failed code:" +
response.getJobResultList().get(0).getCode() +
                                   " message:" + response.getJobResu
ltList().get(0).getMessage());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
```

```
    }
}
```

3.3.13 新增媒体工作流

您可以对MPS服务提供的活动（例如：转码、截图等活动）进行组装，拓扑结构如下。

Topology类

```
package com.aliyun.mts;
import com.alibaba.fastjson.annotation.JSONField;
import java.util.List;
import java.util.Map;
public class Topology {
    @JSONField(name = "Activities")
    private Map<String, Activity> activities;
    @JSONField(name = "Dependencies")
    private Map<String, List<String>> dependencies;
    public Map<String, List<String>> dependencies() {
        return this.getDependencies();
    }
    public Topology() {
    }
    public Topology(Map<String, Activity> activities, Map<String, List
<String>> dependencies) {
        this.setActivities(activities);
        this.setDependencies(dependencies);
    }
    public Map<String, Activity> getActivities() {
        return activities;
    }
    public Map<String, List<String>> getDependencies() {
        return dependencies;
    }
    public void setActivities(Map<String, Activity> activities) {
        this.activities = activities;
    }
    public void setDependencies(Map<String, List<String>> dependencies
) {
        this.dependencies = dependencies;
    }
}
```

ActivityType枚举类

```
package com.aliyun.mts;
/**
 * Created by zhongyizengzy on 18/3/22.
 */
public enum ActivityType {
    Start, Transcode, Snapshot, MediaInfo, Analysis, Cover, Summary
, Censor, Report, UploadVerify, GenerateMasterPlayList, AudioGroup,
SubtitleGroup, PackageConfig
}
```

Activity类

```
package com.aliyun.mts;
import com.alibaba.fastjson.annotation.JSONField;
import java.util.Map;
```

```
public class Activity {
    @JSONField(name = "Type")
    private String type;
    @JSONField(name = "Parameters")
    private Map<String, String> parameters;
    public Activity() {
    }
    public Map<String, String> parameters() {
        return this.getParameters();
    }
    public Activity(String type, Map<String, String> parameters) {
        this.setType(type);
        this.setParameters(parameters);
    }
    public String getType() {
        return type;
    }
    public Map<String, String> getParameters() {
        return parameters;
    }
    public void setType(String type) {
        this.type = type;
    }
    public void setParameters(Map<String, String> parameters) {
        this.parameters = parameters;
    }
}
```

AddMediaWorkflow类

```
package com.aliyun.mts;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.AddMediaWorkflowRequest;
import com.aliyuncs.mts.model.v20140618.AddMediaWorkflowResponse;
import com.aliyuncs.profile.DefaultProfile;
import org.apache.commons.lang.exception.ExceptionUtils;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
public class AddMediaWorkflow {
    //Step 1 .set region: cn-hangzhou、cn-shenzhen、cn-shanghai、cn-
beijing
    private static final String REGION = "cn-shenzhen";
    private static final String OSS_REGION = "oss-cn-shenzhen";
    private static final String mtsEndpoint = "mts." + REGION + ".
aliyuncs.com";
    //Step 2.set accesskey & keySecret
    private static String accessKeyId = "";
    private static String accessKeySecret = "";
    //Step 3.set mps transcoding queue id
    private static String PIPELINE_ID = "38bba54d524448be92d2
77caaa8da118";
    private static DefaultAcsClient aliyunClient;
    static {
        try {
```

```
            DefaultProfile.addEndpoint(REGION, REGION, "Mts",
mtsEndpoint);
        } catch (ClientException e) {
            System.out.print(ExceptionUtils.getStackTrace(e));
            System.exit(1);
        }
        aliyunClient = new DefaultAcsClient(DefaultProfile.getProfile(
REGION, accessKeyId, accessKeySecret));
    }
    public static void main(String[] args) throws ClientException {
        AddMediaWorkflowRequest request = new AddMediaWorkflowRequest
();
        request.setName("Sequential-workflow");
        Topology topology = new Topology();
        HashMap<String, Activity> activities = new HashMap<String,
Activity>();
        Activity startNode = new Activity();
        startNode.setType(ActivityType.Start.name());
        HashMap<String, String> startNodeParameters = new HashMap<
String, String>();
        JSONObject inputFile = new JSONObject();
        inputFile.put("Bucket", "mtb-sz-in");
        inputFile.put("Location", OSS_REGION);
        inputFile.put("ObjectPrefix", "media/");
        startNodeParameters.put("InputFile", inputFile.toString());
        startNodeParameters.put("PipelineId", PIPELINE_ID);
        startNode.setParameters(startNodeParameters);
        activities.put("startNode", startNode);
        Activity transcode = new Activity();
        transcode.setType(ActivityType.Transcode.name());
        HashMap<String, String> transcodingParameters = new HashMap<
String, String>();
        JSONArray outputs = new JSONArray();
        JSONObject output = new JSONObject();
        try {
            output.put("OutputObject", URLEncoder.encode("transcode/{
ObjectPrefix}/{FileName}.{ExtName}", "UTF-8"));
        } catch (UnsupportedEncodingException e) {
            System.exit(1);
        }
        output.put("TemplateId", "S00000001-000070");
        outputs.add(output);
        transcodingParameters.put("Outputs", outputs.toJSONString());
        transcodingParameters.put("OutputBucket", "mtb-sz-out");
        transcodingParameters.put("OutputLocation", OSS_REGION);
        transcode.setParameters(transcodingParameters);
        activities.put("transcodingNode", transcode);
        Activity report = new Activity();
        report.setType(ActivityType.Report.name());
        HashMap<String, String> reportParameters = new HashMap<String
, String>();
        report.setParameters(reportParameters);
        activities.put("reportNode", report);
        topology.setActivities(activities);
        HashMap<String, List<String>> dependencies = new HashMap<
String, List<String>>();
        dependencies.put("startNode", Arrays.asList("transcodingNode
"));
        dependencies.put("transcodingNode", Arrays.asList("reportNode
"));
        dependencies.put("reportNode", new ArrayList<String>());
        topology.setDependencies(dependencies);
        request.setTopology(JSONObject.toJSONString(topology));
        try {
```

```
            AddMediaWorkflowResponse response = aliyunClient.
getAcsResponse(request);
            System.out.println(JSONObject.toJSONString(response));
        } catch (ServerException e) {
            System.out.println("Code:" + e.getErrCode() + " Msg:" + e.
getMessage());
        } catch (ClientException e) {
            System.out.println("Code:" + e.getErrCode() + " Msg:" + e.
getMessage());
        }
    }
}
```

3.3.14 拼接-开板和尾板

1. **创建AcsClient实例。**

```
DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
IAcsClient client = new DefaultAcsClient(profile);
```

2. **创建request，并设置参数。**

```
SubmitJobsRequest request = new SubmitJobsRequest();
```

3. **设置转码参数。**

   - **Input**

```
JSONObject input = new JSONObject();
   input.put("Location", ossLocation);
   input.put("Bucket", ossBucket);
   try {
       input.put("Object", URLEncoder.encode(headObject, "utf-8
"));
   } catch (UnsupportedEncodingException e) {
       throw new RuntimeException("input URL encode failed");
   }
   request.setInput(input.toJSONString());
```

   - **Output**

```
String outputOSSObject;
   try {
       outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-8
");
   } catch (UnsupportedEncodingException e) {
       throw new RuntimeException("output URL encode failed");
   }
   JSONObject output = new JSONObject();
   output.put("OutputObject", outputOSSObject);
   // Ouput->TemplateId
   output.put("TemplateId", templateId);
```

     - **Video**

```
JSONObject video = new JSONObject();
```

```
 video.put("Width", "1280");
 video.put("Height", "720");
 output.put("Video", video.toJSONString());
```

- OpeningList

```
JSONObject openingVideo = new JSONObject();
String openingVideoURL;
try {
    openingVideoURL = String.format(
                                  "http://%s.%s.aliyuncs.com/%s",
                                  ossBucket,
                                  ossLocation,
                                  URLEncoder.encode(headObject, "
utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeVideoURL encode failed");
}
openingVideo.put("OpenUrl", openingVideoURL);
openingVideo.put("Width", "640");
openingVideo.put("Start", "2");
JSONArray openingVideoList = new JSONArray();
openingVideoList.add(openingVideo);
output.put("OpeningList", openingVideoList.toJSONString());
```

- TailSlateList

```
JSONObject tailSlateVideo = new JSONObject();
String tailSlateVideoURL;
try {
    tailSlateVideoURL = String.format(
                                  "http://%s.%s.aliyuncs.com/%s",
                                  ossBucket,
                                  ossLocation,
                                  URLEncoder.encode(tailObject, "
utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("mergeTailURL encode failed");
}
tailSlateVideo.put("TailUrl", tailSlateVideoURL);
tailSlateVideo.put("Width", "640");
tailSlateVideo.put("BlendDuration", "3");
tailSlateVideo.put("BgColor", "Black");
JSONArray tailSlateVideoList = new JSONArray();
tailSlateVideoList.add(tailSlateVideo);
output.put("TailSlateList", tailSlateVideoList.toJSONString());
```

4. 发起API请求并显示返回值。

```
SubmitJobsResponse response;
response = client.getAcsResponse(request);
System.out.println("RequestId is:"+response.getRequestId());
if (response.getJobResultList().get(0).getSuccess()) {
    System.out.println("JobId is:" + response.getJobResultList().get
(0).getJob().getJobId());
} else {
    System.out.println("SubmitJobs Failed code:" + response.
getJobResultList().get(0).getCode() +
                       " message:" + response.getJobResultList().get
(0).getMessage());
```

```
}
```

完整代码

```
package com.aliyun.mts;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.mts.model.v20140618.*;
public class OpenTail {
    private static String accessKeyId = "xxx";
    private static String accessKeySecret = "xxx";
    private static String mpsRegionId = "cn-hangzhou";
    private static String pipelineId = "xxx";
    private static String templateId = "S00000001-200030";
    private static String ossLocation = "oss-cn-hangzhou";
    private static String ossBucket = "xxx";
    private static String ossInputObject = "input.mp4";
    private static String ossOutputObject = "output.mp4";
    private static String headObject = "head.mp4";
    private static String tailObject = "tail.mp4";
    public static void main(String[] args) {
        // DefaultAcsClient
        DefaultProfile profile = DefaultProfile.getProfile(
                mpsRegionId,      // Region ID
                accessKeyId,      // AccessKey ID
                accessKeySecret); // Access Key Secret
        IAcsClient client = new DefaultAcsClient(profile);
        // request
        SubmitJobsRequest request = new SubmitJobsRequest();
        // Input
        JSONObject input = new JSONObject();
        input.put("Location", ossLocation);
        input.put("Bucket", ossBucket);
        try {
            input.put("Object", URLEncoder.encode(ossInputObject, "utf
-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("input URL encode failed");
        }
        request.setInput(input.toJSONString());
        // Output
        String outputOSSObject;
        try {
            outputOSSObject = URLEncoder.encode(ossOutputObject, "utf-
8");
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("output URL encode failed");
        }
        JSONObject output = new JSONObject();
        output.put("OutputObject", outputOSSObject);
        // Ouput->TemplateId
        output.put("TemplateId", templateId);
        // Output->OpeningList
        JSONObject openingVideo = new JSONObject();
        String openingVideoURL;
        try {
```

```
            openingVideoURL = String.format(
                                        "http://%s.%s.aliyuncs.com/%s
",
                                        ossBucket,
                                        ossLocation,
                                        URLEncoder.encode(headObject,
 "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("mergeVideoURL encode failed");
        }
        openingVideo.put("OpenUrl", openingVideoURL);
        openingVideo.put("Width", "640");
        openingVideo.put("Start", "2");
        JSONArray openingVideoList = new JSONArray();
        openingVideoList.add(openingVideo);
        output.put("OpeningList", openingVideoList.toJSONString());
        // Output->TailSlateList
        JSONObject tailSlateVideo = new JSONObject();
        String tailSlateVideoURL;
        try {
            tailSlateVideoURL = String.format(
                                        "http://%s.%s.aliyuncs.com/%s
",
                                        ossBucket,
                                        ossLocation,
                                        URLEncoder.encode(tailObject,
 "utf-8"));
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("mergeTailURL encode failed");
        }
        tailSlateVideo.put("TailUrl", tailSlateVideoURL);
        tailSlateVideo.put("Width", "640");
        tailSlateVideo.put("BlendDuration", "3");
        tailSlateVideo.put("BgColor", "Black");
        JSONArray tailSlateVideoList = new JSONArray();
        tailSlateVideoList.add(tailSlateVideo);
        output.put("TailSlateList", tailSlateVideoList.toJSONString
());
        // Outputs
        JSONArray outputs = new JSONArray();
        outputs.add(output);
        request.setOutputs(outputs.toJSONString());
        request.setOutputBucket(ossBucket);
        request.setOutputLocation(ossLocation);
        // PipelineId
        request.setPipelineId(pipelineId);
        // call api
        SubmitJobsResponse response;
        try {
            response = client.getAcsResponse(request);
            System.out.println("RequestId is:"+response.getRequestId
());
            if (response.getJobResultList().get(0).getSuccess()) {
                System.out.println("JobId is:" + response.getJobResu
ltList().get(0).getJob().getJobId());
            } else {
                System.out.println("SubmitJobs Failed code:" +
response.getJobResultList().get(0).getCode() +
                                   " message:" + response.getJobResu
ltList().get(0).getMessage());
            }
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
```

```
            e.printStackTrace();
        }
    }
}
```

3.4 Python SDK

3.4.1 前言

媒体处理的Python SDK基于阿里云Python SDK，本文介绍阿里云Python SDK的一些基本知识。

- 阿里云*Python SDK*快速入门
- 阿里云*Python SDK*使用手册
- 阿里云*Python SDK GitHub*仓库

了解基本知识后，您可以具体进行媒体处理Python SDK的安装了。详情参见 媒体处理 > *SDK*参考 > 媒体转码*SDK* > *Python SDK* > 安装。

3.4.2 安装

本文介绍阿里云Python SDK推荐的pip安装方式。

- Python 2.x

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

- Python 3.x

```
```
pip install aliyun-python-sdk-core-v3
pip install aliyun-python-sdk-mts
```
```

3.4.3 快速入门

1. 创建AcsClient实例。

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. 创建request，并设置参数。

```
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
```

3. 发起API请求并显示返回值。

```
response_str = client.do_action_with_exception(request)
```

```
response = json.loads(response_str)
print 'PipelineName is:', response['PipelineList']['Pipeline'][0]['
Name']
print 'PipelineId is:', response['PipelineList']['Pipeline'][0]['Id
']
```

完整代码

```
# -*- coding: utf8 -*-
import json
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SearchPipelineRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_ids = ['cn-hangzhou', 'cn-beijing', 'cn-shenzhen', 'cn-
shanghai',
                      'cn-hongkong', 'us-west-1', 'ap-southeast-1',
                      'ap-northeast-1', 'eu-central-1', 'ap-south-1']
for mps_region_id in mps_region_ids:
    print 'region is:', mps_region_id
    # 创建AcsClient实例
    client = AcsClient(access_key_id, access_key_secret, mps_region_id
)
    # 创建request，并设置参数
    request = SearchPipelineRequest.SearchPipelineRequest()
    request.set_accept_format('json')
    # 发起API请求并显示返回值
    response_str = client.do_action_with_exception(request)
    response = json.loads(response_str)
    print 'PipelineName is:', response['PipelineList']['Pipeline'][0]
['Name']
    print 'PipelineId is:', response['PipelineList']['Pipeline'][0]['
Id']
```

3.4.4 转码

1. **创建AcsClient实例。**

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. **创建request，并设置参数。**

```
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
```

3. **设置转码参数。**

   - **Input**

```
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(oss_input_object) }
```

```
request.set_Input(json.dumps(job_input))
```

- Output

```
output = {'OutputObject': quote(oss_output_object)}
```

  - Container

```
output['Container'] = {'Format': 'mp4'}
```

  - Video

```
output['Video'] = {'Codec': 'H.264',
                   'Bitrate': 1500,
                   'Width': 1280,
                   'Fps': 25}
```

  - Audio

```
output['Audio'] = {'Codec': 'AAC',
                   'Bitrate': 128,
                   'Channels': 2,
                   'Samplerate': 44100}
```

  - TemplateId

```
output['TemplateId'] = template_id
```

- PipelineId

```
request.set_PipelineId(pipeline_id)
```

4. 发起API请求并显示返回值。

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
print 'JobId is:', response['JobResultList']['JobResult'][0]['Job']
['JobId']
else:
print ('SubmitJobs Failed code:',
      response['JobResultList']['JobResult'][0]['Code'],
      ' message:',
      response['JobResultList']['JobResult'][0]['Message'])
```

完整代码

```
# -*- coding: utf8 -*-
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
template_id = 'S00000001-200010'
```

```
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output.mp4'
# 创建AcsClient实例
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# 创建request，并设置参数
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(oss_input_object) }
request.set_Input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Ouput->Container
output['Container'] = {'Format': 'mp4'}
# Ouput->Video
output['Video'] = {'Codec': 'H.264',
                   'Bitrate': 1500,
                   'Width': 1280,
                   'Fps': 25}
# Ouput->Audio
output['Audio'] = {'Codec': 'AAC',
                   'Bitrate': 128,
                   'Channels': 2,
                   'Samplerate': 44100}
# Ouput->TemplateId
output['TemplateId'] = template_id
outputs = [output]
request.set_Outputs(json.dumps(outputs))
request.set_OutputBucket(oss_bucket)
request.set_OutputLocation(oss_location)
# PipelineId
request.set_PipelineId(pipeline_id)
# 发起API请求并显示返回值
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job
']['JobId']
else:
    print ('SubmitJobs Failed code:',
           response['JobResultList']['JobResult'][0]['Code'],
           ' message:',
           response['JobResultList']['JobResult'][0]['Message'])
```

3.4.5 水印

1. 创建AcsClient实例。

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. 创建request，并设置参数。

```
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
```

3. 设置水印参数。

- **图片水印**

```
image_watermark_input = {
                    'Location': oss_location,
                    'Bucket': oss_bucket,
                    'Object': quote(image_watermark_object)
                    }
 image_watermark = {
             'WaterMarkTemplateId': watermark_template_id,
             'Type': 'Image',
             'InputFile': image_watermark_input,
             'ReferPos': 'TopRight',
             'Width': 0.05,
             'Dx': 0,
             'Dy': 0
            }
```

- **文字水印**

```
text_config = {
          'Content': '5rWL6K+V5paH5a2X5rC05Y2w',
          'FontName': 'SimSun',
          'FontSize': 16,
          'FontColor': 'Red',
          'FontAlpha': 0.5,
          'Top': 10,
          'Left': 10
         }
 text_watermark = {
             'WaterMarkTemplateId': watermark_template_id,
             'Type': 'Text',
             'TextWaterMark': text_config
            }
```

- **视频水印**

```
video_watermark_input = {
                    'Location': oss_location,
                    'Bucket': oss_bucket,
                    'Object': quote(video_watermark_object)
                    }
 video_watermark = {
             'WaterMarkTemplateId': watermark_template_id,
             'Type': 'Image',
             'InputFile': video_watermark_input,
             'ReferPos': 'BottomLeft',
             'Height': 240,
             'Dx': 0,
             'Dy': 0
            }
```

4. **发起API请求并显示返回值。**

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['
Job']['JobId']
else:
    print ('SubmitJobs Failed code:',
```

```
            response['JobResultList']['JobResult'][0]['Code'],
            ' message:',
            response['JobResultList']['JobResult'][0]['Message'])
```

完整代码

```
# -*- coding: utf8 -*-
from pprint import pprint
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
watermark_template_id = 'xxx'
template_id = 'S00000001-200030'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output.mp4'
image_watermark_object = 'logo.png'
video_watermark_object = 'logo.mov'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(oss_input_object) }
request.set_Input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Ouput->TemplateId
output['TemplateId'] = template_id
## Image Watermark
image_watermark_input = {'Location': oss_location,
                         'Bucket': oss_bucket,
                         'Object': quote(image_watermark_object) }
image_watermark = {
                  'WaterMarkTemplateId': watermark_template_id,
                  'Type': 'Image',
                  'InputFile': image_watermark_input,
                  'ReferPos': 'TopRight',
                  'Width': 0.05,
                  'Dx': 0,
                  'Dy': 0
                 }
## Text Watermark
text_config = {
                'Content': '5rWL6K+V5paH5a2X5rC05Y2w',
                'FontName': 'SimSun',
                'FontSize': 16,
                'FontColor': 'Red',
                'FontAlpha': 0.5,
                'Top': 10,
                'Left': 10
               }
text_watermark = {
                  'WaterMarkTemplateId': watermark_template_id,
```

```
                    'Type': 'Text',
                    'TextWaterMark': text_config
                   }
## Video Watermark
video_watermark_input = {'Location': oss_location,
                         'Bucket': oss_bucket,
                         'Object': quote(video_watermark_object) }
video_watermark = {
                    'WaterMarkTemplateId': watermark_template_id,
                    'Type': 'Image',
                    'InputFile': video_watermark_input,
                    'ReferPos': 'BottomLeft',
                    'Height': 240,
                    'Dx': 0,
                    'Dy': 0
                   }
# Output->Watermarks
watermarks = [image_watermark, text_watermark, video_watermark]
output['WaterMarks'] = watermarks
# Outputs
outputs = [output]
request.set_Outputs(json.dumps(outputs))
request.set_OutputBucket(oss_bucket)
request.set_OutputLocation(oss_location)
# PipelineId
request.set_PipelineId(pipeline_id)
# call api
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job
']['JobId']
else:
    print ('SubmitJobs Failed code:',
           response['JobResultList']['JobResult'][0]['Code'],
           ' message:',
           response['JobResultList']['JobResult'][0]['Message'])
```

3.4.6 截图

1. **创建AcsClient实例。**

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. **创建request，并设置参数。**

```
request = SubmitSnapshotJobRequest.SubmitSnapshotJobRequest()
request.set_accept_format('json')
```

3. **设置截图参数。**

   - **Input**

```
job_input = {'Location': oss_location,
        'Bucket': oss_bucket,
        'Object': quote(head_object) }
 request.set_Input(json.dumps(job_input))
```

   - **SnapshotConfig**

- **OutputFile**

```
job_output = {'Location': oss_location,
      'Bucket': oss_bucket,
      'Object': quote(oss_output_object) }
snapshot_config = {'OutputFile': job_output}
```

- **Time**

```
snapshot_config['Time'] = 2
```

- **Interval/Num**

```
snapshot_config['Interval'] = 2
snapshot_config['Num'] = 3
```

- **Width/Height**

```
snapshot_config['Height'] = 360
```

4. 发起API请求并显示返回值。

```
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print response
print 'RequestId is:', response['RequestId']
print 'JobId is:', response['SnapshotJob']['Id']
print 'http://%s.%s.aliyuncs.com/output_00001.jpg' % (oss_bucket,
oss_location)
print 'http://%s.%s.aliyuncs.com/output_00002.jpg' % (oss_bucket,
oss_location)
print 'http://%s.%s.aliyuncs.com/output_00003.jpg' % (oss_bucket,
oss_location)
```

完整代码

```
# -*- coding: utf8 -*-
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitSnapshotJobRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output_{Count}.jpg'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitSnapshotJobRequest.SubmitSnapshotJobRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(oss_input_object) }
request.set_Input(json.dumps(job_input))
```

```
# SnapshotConfig->OutputFile
job_output = {'Location': oss_location,
              'Bucket': oss_bucket,
              'Object': quote(oss_output_object) }
snapshot_config = {'OutputFile': job_output}
# SnapshotConfig->Time
snapshot_config['Time'] = 2
# SnapshotConfig->Interval/Num
snapshot_config['Interval'] = 2
snapshot_config['Num'] = 3
# SnapshotConfig->Width/Height
snapshot_config['Height'] = 360
# SnapshotConfig
request.set_SnapshotConfig(json.dumps(snapshot_config))
# PipelineId
request.set_PipelineId(pipeline_id)
# call api
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print response
print 'RequestId is:', response['RequestId']
print 'JobId is:', response['SnapshotJob']['Id']
print 'http://%s.%s.aliyuncs.com/output_00001.jpg' % (oss_bucket,
oss_location)
print 'http://%s.%s.aliyuncs.com/output_00002.jpg' % (oss_bucket,
oss_location)
print 'http://%s.%s.aliyuncs.com/output_00003.jpg' % (oss_bucket,
oss_location)
```

3.4.7 查询媒体-使用OSS文件地址

如果您不知道媒体ID（直播走工作流转点播），可以通过媒体的输入地址进行媒体信息的查询，接口为 QueryMediaListByURL。

```
import json
from aliyunsdkmts.request.v20140618 import QueryMediaListByURLRequest
from aliyunsdkcore import client
import urllib
region = '<region>'
access_key_id = '<accessKeyId>'
access_key_secret = '<accessKeySecret>'
def queryMediaListByURL():
    global client
    client = client.AcsClient(access_key_id, access_key_secret, region
)
    request = QueryMediaListByURLRequest.QueryMediaListByURLRequest()
    ossDomain = 'http://<input-bucket>.<region>.aliyuncs.com/';
    #对ossObject进行encode
    ossObject = encodeByRFC3986("test/笑傲江湖.mp4")
    request.set_FileURLs(ossDomain + ossObject)
    response = client.do_action_with_exception(request);
    json_response = json.loads(response)
    print json_response
def encodeByRFC3986(ossObject):
    return urllib.quote(ossObject)
if __name__ == "__main__":
```

```
    queryMediaListByURL()
```

3.4.8 创建HLS标准加密工作流

简介

示例调用API，创建HLS标准加密工作流。创建HLS标准加密工作流到播放加密视频的完整步骤，参见 *HLS*的加密与播放。MPS SDK，详情参见 安装。

示例代码

```
import json
from aliyunsdkmts.request.v20140618 import AddMediaWorkflowRequest
from aliyunsdkcore import client
REGION_ID = '<region>'
ACCESS_KEY_ID = '<accessKeyId>'
ACCESS_KEY_SECRET = '<accessKeySecret>'
PIPELINE_ID = "<PipelineId>"
TEMPLATE_ID = "S00000001-100020" #转码模版ID，m3u8模版，按需配置
OSS_LOCATION = "<OssLocation>"
INPUT_BUCKET = "<InputBucket>"
INPUT_PATH = "<InputPath>" #如 "HLS-Encryption"
OUTPUT_BUCKET = "<OutputBucket>"
ENCRYPTION_TYPE = "hls-aes-128"
HLS_KEY_URI = "<解密密钥的URI>" #如http://decrypt.testdomain.com
ACT_START = "Act-Start"
ACT_ENCRYPTION = "Act-HLS-Encryption"
ACT_REPORT = "Act-Report"
def addMediaWorkflow():
    global client
    client = client.AcsClient(ACCESS_KEY_ID, ACCESS_KEY_SECRET,
REGION_ID)
    request = AddMediaWorkflowRequest.AddMediaWorkflowRequest()
    request.set_Topology(buildWorkflowTopology())
    request.set_Name("HLS加密工作流py")
    response = client.do_action_with_exception(request)
    print json.loads(response)
def buildWorkflowTopology():
    workflow = {}
    workflow["Activities"] = buildActivities()
    workflow["Dependencies"] = buildDependencies()
    print json.dumps(workflow)
    return json.dumps(workflow)
def buildActivities():
    activities = {}
    activities[ACT_START] = buildStartActivity()
    activities[ACT_ENCRYPTION] = buildTranscodeActivity()
    activities[ACT_REPORT] = buildReportActivity()
    return activities
def buildStartActivity():
    startActivity = {}
    startActivity["Name"] = ACT_START
    startActivity["Type"] = "Start"
    startActivity["Parameters"] = buildStartParameters()
    return startActivity
def buildStartParameters():
    startParameters = {}
    startParameters["PipelineId"] = PIPELINE_ID
    startParameters["InputFile"] = buildInputFile();
    return startParameters
```

```
def buildInputFile():
    inputFile = {}
    inputFile["Bucket"] = INPUT_BUCKET
    inputFile["Location"] = OSS_LOCATION
    inputFile["ObjectPrefix"] = INPUT_PATH
    return inputFile
def buildTranscodeActivity():
    transcodeActivity = {}
    transcodeActivity["Name"] = ACT_ENCRYPTION
    transcodeActivity["Type"] = "Transcode"
    transcodeActivity["Parameters"] = buildTranscodeParameters()
    return transcodeActivity
def buildTranscodeParameters():
    transcodeParameters = {}
    transcodeParameters["OutputBucket"] = OUTPUT_BUCKET
    transcodeParameters["OutputLocation"] = OSS_LOCATION
    transcodeParameters["Outputs"] = buildOutputsConfig()
    return transcodeParameters
def buildOutputsConfig():
    outputs = []
    output = {}
    output["ObjectRegex"] = ACT_ENCRYPTION + "/{RunId}/{FileName}"
    output["TemplateId"] = TEMPLATE_ID
    output["Encryption"] = buildEncryption()
    outputs.append(output)
    return outputs
def buildEncryption():
    encryption = {}
    encryption["Type"] = ENCRYPTION_TYPE
    encryption["KeyUri"] = HLS_KEY_URI
    return encryption
def buildReportActivity():
    reportActivity = {}
    reportActivity["Name"] = ACT_REPORT
    reportActivity["Parameters"] = buildReportParameters()
    reportActivity["Type"] = "Report"
    return reportActivity
def buildReportParameters():
    parameters = {}
    parameters["PublishType"] = "Auto"
    return parameters
def buildDependencies():
    dependencies ={}
    subActivityOfStart = [ACT_ENCRYPTION]
    dependencies[ACT_START] = subActivityOfStart
    subActivityOfTranscode = [ACT_REPORT]
    dependencies[ACT_ENCRYPTION] = subActivityOfTranscode
    dependencies[ACT_REPORT] = []
    return dependencies
if __name__ == "__main__":
    addMediaWorkflow()
```

3.4.9 拼接和简单剪辑

1. **创建AcsClient实例。**

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. **创建request，并设置参数。**

```
request = SubmitJobsRequest.SubmitJobsRequest()
```

```
request.set_accept_format('json')
```

3. 设置转码参数。

- Input

```
job_input = {'Location': oss_location,
        'Bucket': oss_bucket,
        'Object': quote(head_object) }
 request.set_Input(json.dumps(job_input))
```

- Output

```
output = {'OutputObject': quote(oss_output_object)}
```

  - Video

```
output['Video'] = {'Width': 1280,
                   'Height': 720}
```

  - MergeList

```
merge_video = {'MergeURL': 'http://%s.%s.aliyuncs.com/%s'%(
oss_bucket, oss_location, quote(oss_input_object))}
 merge_tail = {'MergeURL': 'http://%s.%s.aliyuncs.com/%s'%(
oss_bucket, oss_location, quote(tail_object))}
 output['MergeList'] = [merge_video, merge_tail]
```

4. 发起API请求并显示返回值。

```
response_str = client.do_action_with_exception(request)
  response = json.loads(response_str)
  print 'RequestId is:', response['RequestId']
  if response['JobResultList']['JobResult'][0]['Success']:
      print 'JobId is:', response['JobResultList']['JobResult'][0]['
Job']['JobId']
  else:
      print ('SubmitJobs Failed code:',
             response['JobResultList']['JobResult'][0]['Code'],
             'message:',
             response['JobResultList']['JobResult'][0]['Message'])
```

完整代码

```
# -*- coding: utf8 -*-
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
template_id = 'S00000001-200030'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
oss_output_object = 'output.mp4'
head_object = 'head.mp4'
```

```
tail_object = 'tail.mp4'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(head_object) }
request.set_Input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Ouput->TemplateId
output['TemplateId'] = template_id
# Ouput->Video
output['Video'] = {'Width': 1280,
                   'Height': 720}
# Output->MergeList
merge_video = {'MergeURL': 'http://%s.%s.aliyuncs.com/%s'%(oss_bucket
, oss_location, quote(oss_input_object))}
merge_tail = {'MergeURL': 'http://%s.%s.aliyuncs.com/%s'%(oss_bucket,
oss_location, quote(tail_object))}
output['MergeList'] = [merge_video, merge_tail]
# Outputs
outputs = [output]
request.set_Outputs(json.dumps(outputs))
request.set_OutputBucket(oss_bucket)
request.set_OutputLocation(oss_location)
# PipelineId
request.set_PipelineId(pipeline_id)
# call api
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job
']['JobId']
else:
    print ('SubmitJobs Failed code:',
           response['JobResultList']['JobResult'][0]['Code'],
           ' message:',
           response['JobResultList']['JobResult'][0]['Message'])
```

3.4.10 拼接-开板和尾板

1. **创建AcsClient实例。**

```
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
```

2. **创建request，并设置参数。**

```
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
```

3. **设置转码参数。**

   - **Input**

```
job_input = {'Location': oss_location,
```

```
        'Bucket': oss_bucket,
        'Object': quote(head_object) }
  request.set_Input(json.dumps(job_input))
```

- Output

```
output = {'OutputObject': quote(oss_output_object)}
```

  - Video

```
output['Video'] = {'Width': 1280,
                   'Height': 720}
```

  - OpeningList

```
opening_video = {'OpenUrl': 'http://%s.%s.aliyuncs.com/%s'%(
oss_bucket, oss_location, quote(head_object)),
          'Width': 640,
          'Start': 2}
output['OpeningList'] = [opening_video]
```

  - TailSlateList

```
tailslate_video = {'TailUrl': 'http://%s.%s.aliyuncs.com/%s'%(
oss_bucket, oss_location, quote(tail_object)),
           'Width': 640,
           'BlendDuration': 3,
           'BgColor': 'Black'}
output['TailSlateList'] = [tailslate_video]
```

4. 发起API请求并显示返回值。

```
response_str = client.do_action_with_exception(request)
  response = json.loads(response_str)
  print 'RequestId is:', response['RequestId']
  if response['JobResultList']['JobResult'][0]['Success']:
      print 'JobId is:', response['JobResultList']['JobResult'][0]['
Job']['JobId']
  else:
      print ('SubmitJobs Failed code:',
             response['JobResultList']['JobResult'][0]['Code'],
             'message:',
             response['JobResultList']['JobResult'][0]['Message'])
```

完整代码

```
# -*- coding: utf8 -*-
import json
from urllib import quote
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SubmitJobsRequest
access_key_id = 'xxx'
access_key_secret = 'xxx'
mps_region_id = 'cn-hangzhou'
pipeline_id = 'xxx'
template_id = 'S00000001-200030'
oss_location = 'oss-cn-hangzhou'
oss_bucket = 'xxx'
oss_input_object = 'input.mp4'
```

```
oss_output_object = 'output.mp4'
head_object = 'head.mp4'
tail_object = 'tail.mp4'
# AcsClient
client = AcsClient(access_key_id, access_key_secret, mps_region_id);
# request
request = SubmitJobsRequest.SubmitJobsRequest()
request.set_accept_format('json')
# Input
job_input = {'Location': oss_location,
             'Bucket': oss_bucket,
             'Object': quote(oss_input_object) }
request.set_Input(json.dumps(job_input))
# Output
output = {'OutputObject': quote(oss_output_object)}
# Ouput->TemplateId
output['TemplateId'] = template_id
# Output->OpeningList
opening_video = {'OpenUrl': 'http://%s.%s.aliyuncs.com/%s'%(oss_bucket
, oss_location, quote(head_object)),
                 'Width': 640,
                 'Start': 2}
output['OpeningList'] = [opening_video]
# Output->TailSlateList
tailslate_video = {'TailUrl': 'http://%s.%s.aliyuncs.com/%s'%(
oss_bucket, oss_location, quote(tail_object)),
                   'Width': 640,
                   'BlendDuration': 3,
                   'BgColor': 'Black'}
output['TailSlateList'] = [tailslate_video]
# Outputs
outputs = [output]
request.set_Outputs(json.dumps(outputs))
request.set_OutputBucket(oss_bucket)
request.set_OutputLocation(oss_location)
# PipelineId
request.set_PipelineId(pipeline_id)
# call api
response_str = client.do_action_with_exception(request)
response = json.loads(response_str)
print 'RequestId is:', response['RequestId']
if response['JobResultList']['JobResult'][0]['Success']:
    print 'JobId is:', response['JobResultList']['JobResult'][0]['Job
']['JobId']
else:
    print ('SubmitJobs Failed code:',
           response['JobResultList']['JobResult'][0]['Code'],
           ' message:',
           response['JobResultList']['JobResult'][0]['Message'])
```

3.4.11 添加媒体

新增媒体，添加视频文件到媒体库，可指定工作流触发处理此视频。

```
import json
from aliyunsdkcore.acs_exception.exceptions import ServerException,
ClientException
from aliyunsdkmts.request.v20140618 import AddMediaRequest
from aliyunsdkcore import client
import urllib
import thread
# Step 1 set region
```

```
REGION = "cn-shenzhen";
mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
accessKeyId = "";
accessKeySecret = "";
cli = client.AcsClient(accessKeyId, accessKeySecret, REGION)
def addMeida():
    request = AddMediaRequest.AddMediaRequest()
    request.set_FileURL("http://mtb-sz-in.oss-cn-shenzhen.aliyuncs.com
/media/r180-ABC.mp4")
    request.set_MediaWorkflowId("829bed0300994057a49e4f16de957e34")
    try:
        response = cli.do_action_with_exception(request)
        json_response = json.loads(response)
        print json.dumps(json_response)
    except ServerException, e:
        print e.get_error_code(), e.get_error_msg()
    except ClientException, e:
        print e.get_error_code(), e.get_error_msg()
def encodeByRFC3986(ossObject):
    return urllib.quote(ossObject)
if __name__ == "__main__":
    addMeida()
```

3.4.12 管道管理

在开通服务时，系统会自动创建一个用户管道。您可以通过一系列接口管理管道（pipeline）。例如：SearchPipeline、QueryPipelineList、UpdatePipeline。

搜索管道

可以直接通过SearchPipeline**接口搜索管道信息。**

```
import json
    from aliyunsdkmts.request.v20140618 import SearchPipelineRequest
    from aliyunsdkcore import client
    region = '<region>'
    access_key_id = '<access_key_id>'
    access_key_secret = '<access_key_secret>'
    client = client.AcsClient(access_key_id, access_key_secret, region
)
    request = SearchPipelineRequest.SearchPipelineRequest()
    response = client.do_action_with_exception(request);
    json_response = json.loads(response)
    pipelines = json_response['PipelineList']['Pipeline']
    for pipeline in pipelines:
        print 'pipeline id:' + pipeline['Id'] + ', name:' + pipeline['
Name'] + ', state:' + pipeline['State']
```

查询管道

如果已经知道pipelineId**，可以通过**pipelineId**调用**QueryPipelineList**接口查询管道信息。**

```
import json
    from aliyunsdkmts.request.v20140618 import QueryPipelineListReq
uest
    from aliyunsdkcore import client
    region = '<region>'
    access_key_id = '<access_key_id>'
```

```
    access_key_secret = '<access_key_secret>'
    pipeline_id = '<pipeline_id>'
    client = client.AcsClient(access_key_id, access_key_secret, region
)
    request = QueryPipelineListRequest.QueryPipelineListRequest()
    request.set_PipelineIds(pipeline_id)
    response = client.do_action_with_exception(request);
    json_response = json.loads(response)
    pipelines = json_response['PipelineList']['Pipeline']
    for pipeline in pipelines:
        print 'pipeline id:' + pipeline['Id'] + ', name:' + pipeline['
Name'] + ', state:' + pipeline['State']
```

更新管道

通过UpdatePipeline更新管道信息，包括更新管道名称，状态。管道的状态包括Active、Paused。

```
import json
    from aliyunsdkmts.request.v20140618 import SearchPipelineRequest
    from aliyunsdkmts.request.v20140618 import UpdatePipelineRequest
    from aliyunsdkcore import client
    region = '<region>'
    access_key_id = '<access_key_id>'
    access_key_secret = '<access_key_secret>'
    client = client.AcsClient(access_key_id, access_key_secret, region
)
    request = SearchPipelineRequest.SearchPipelineRequest()
    response = client.do_action_with_exception(request);
    json_response = json.loads(response)
    pipeline = json_response['PipelineList']['Pipeline'][0]
    request = UpdatePipelineRequest.UpdatePipelineRequest()
    request.set_PipelineId(pipeline['Id'])
    request.set_Name(pipeline['Name'])
    request.set_State('Paused' if pipeline['State'] == 'Active' else '
Active')
    response = client.do_action_with_exception(request);
    json_response = json.loads(response)
    pipeline = json_response['Pipeline']
    print 'pipeline id:' + pipeline['Id'] + ', name:' + pipeline['Name
'] + ', state:' + pipeline['State']
```

3.4.13 添加媒体工作流

您可以对MPS服务提供的一些活动（转码，截图之类的活动）进行组装，拓扑结构如下。

```
import json
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618 import AddMediaWorkflowRequest
from aliyunsdkcore import client
import urllib
import thread
# Step 1 set region
REGION = "cn-shenzhen";
OSS_REGION = "oss-cn-shenzhen";
mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
accessKeyId = "";
accessKeySecret = "";
# Step 3.set mps transcoding queue id
```

```
PIPELINE_ID = "38bba54d524448be92d277caaa8da118";
cl = client.AcsClient(accessKeyId, accessKeySecret, REGION)
def addMeidaWorkflow():
    request = AddMediaWorkflowRequest.AddMediaWorkflowRequest()
    request.set_Name("Sequential-workflow");
    startActivity = {
        "Type": "Start",
        "Parameters": {
            "InputFile": {
                "Bucket": "mtb-sz-in",
                "Location": OSS_REGION,
                "ObjectPrefix": "media/"
            },
            "PipelineId": PIPELINE_ID
        }
    }
    transcodeActivity = {
        "Type": "Transcode",
        "Parameters": {
            "Outputs": [
                {
                    "OutputObject": encodeByRFC3986("transcode/{
ObjectPrefix}/{FileName}.{ExtName}"),
                    "TemplateId": "S00000001-000070"
                }
            ],
            "OutputLocation": OSS_REGION,
            "OutputBucket": "mtb-sz-out"
        }
    }
    reportActivity = {
        "Type": "Report",
        "Parameters": {
        }
    }
    topology = {
        "Activities": {
            "startNode": startActivity,
            "transcodingNode": transcodeActivity,
            "reportNode": reportActivity
        },
        "Dependencies": {
            "startNode": ["transcodingNode"],
            "transcodingNode": ["reportNode"],
            "reportNode": []
        }
    }
    request.set_Topology(topology)
    try:
        response = json.loads(cl.do_action_with_exception(request))
        print json.dumps(response)
    except ServerException, e:
        print e.get_error_code(), e.get_error_msg()
def encodeByRFC3986(ossObject):
    return urllib.quote(ossObject)
if __name__ == "__main__":
```

```
    addMeidaWorkflow()
```

3.5 PHP SDK

3.5.1 前言

媒体处理的PHP SDK基于阿里云PHP SDK。本文介绍阿里云PHP SDK的一些基本知识。

- 阿里云*PHP SDK*使用手册
- 阿里云*PHP SDK GitHub*仓库

了解基本知识后，就可以具体进行媒体处理PHP SDK的安装了。详情参见 媒体处理 > *SDK*参考 > 媒体转码*SDK* > *PHP SDK* > 安装。

3.5.2 安装

本文介绍阿里云PHP SDK的安装方式。

1. 下载源代码。

```
git clone https://github.com/aliyun/aliyun-openapi-php-sdk.git
```

2. 添加引用。

   假设PHP SDK下载后的路径为`/path/to/aliyun-openapi-php-sdk`

```
require_once '/path/to/aliyun-openapi-php-sdk/aliyun-php-sdk-core/
Config.php';
```

   假设引用项目当前目录下的`aliyun-openapi-php-sdk`

```
require_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php
';
```

3. 自动加载媒体处理SDK。

   编辑文件`aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php`

   找到内容`//config sdk auto load path.`，在后面添加`Autoloader::addAutoloa dPath("aliyun-php-sdk-mts");`

3.5.3 快速入门

1. 创建AcsClient实例。

```
$clientProfile = DefaultProfile::getProfile(
$mps_region_id,                    # 您的 Region ID
$access_key_id,                    # 您的 AccessKey ID
$access_key_secret                 # 您的 AccessKey Secret
);
```

```
$client = new DefaultAcsClient($clientProfile);
```

2. 创建request，并设置参数。

```
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
```

3. 发起API请求并显示返回值。

```
$response = $client->getAcsResponse($request);
print 'PipelineName is:' . $response->{'PipelineList'}->{'Pipeline'}
[0]->{'Name'} . "\n";
print 'PipelineId is:' . $response->{'PipelineList'}->{'Pipeline'}[0
]->{'Id'} . "\n";
```

完整代码

```
<?php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_ids = array('cn-hangzhou', 'cn-beijing', 'cn-shenzhen',
                        'cn-shanghai', 'cn-hongkong', 'us-west-1',
                        'ap-southeast-1', 'ap-northeast-1', 'eu-
central-1',
                        'ap-south-1');
foreach ($mps_region_ids as $mps_region_id) {
    print 'region is:' . $mps_region_id . "\n";
    # 创建DefaultAcsClient实例并初始化
    $clientProfile = DefaultProfile::getProfile(
        $mps_region_id,                  # 您的 Region ID
        $access_key_id,                  # 您的 AccessKey ID
        $access_key_secret               # 您的 AccessKey Secret
    );
    $client = new DefaultAcsClient($clientProfile);
    # 创建API请求并设置参数
    $request = new Mts\SearchPipelineRequest();
    # 发起请求并处理返回
    try {
        $response = $client->getAcsResponse($request);
        print 'PipelineName is:' . $response->{'PipelineList'}->{'
Pipeline'}[0]->{'Name'} . "\n";
        print 'PipelineId is:' . $response->{'PipelineList'}->{'
Pipeline'}[0]->{'Id'} . "\n";
    } catch(ServerException $e) {
        print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
    } catch(ClientException $e) {
        print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
    }
}
```

3.5.4 转码

1. 创建AcsClient实例。

```
$clientProfile = DefaultProfile::getProfile(
$mps_region_id,                  # 您的 Region ID
```

```
$access_key_id,                    # 您的 AccessKey ID
$access_key_secret                 # 您的 AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
```

2. 创建request，并设置参数。

```
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
```

3. 转码参数。

   · Input

```
$input = array('Location' => $oss_location,
               'Bucket' => $oss_bucket,
               'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
```

   · Output

```
$output = array('OutputObject' => urlencode($oss_output_object));
```

   - Container

```
$output['Container'] = array('Format' => 'mp4');
```

   - Video

```
$output['Video'] = array('Codec' =>'H.264',
                         'Bitrate' => 1500,
                         'Width' => 1280,
                         'Fps' => 25);
```

   - Audio

```
$output['Audio'] = array('Codec' => 'AAC',
                         'Bitrate' => 128,
                         'Channels' => 2,
                         'Samplerate' => 44100);
```

   - TemplateId

```
$output['TemplateId'] = $template_id;
```

   · PipelineId

```
$request->setPipelineId($pipeline_id);
```

4. 发起API请求并显示返回值。

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";;
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
print 'JobId is:' .
      $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}->{'
JobId'} . "\n";
} else {
```

```
print 'SubmitJobs Failed code:' .
      $response->{'JobResultList'}->{'JobResult'}[0]->{'Code'} .
      ' message:' .
      $response->{'JobResultList'}->{'JobResult'}[0]->{'Message'} .
 "\n";
}
```

完整代码

```
<?php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$template_id = 'S00000001-200010';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
# 创建DefaultAcsClient实例并初始化
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                    # 您的 Region ID
    $access_key_id,                    # 您的 AccessKey ID
    $access_key_secret                 # 您的 AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# 创建API请求并设置参数
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
               'Bucket' => $oss_bucket,
               'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Ouput->Container
$output['Container'] = array('Format' => 'mp4');
# Ouput->Video
$output['Video'] = array('Codec' =>'H.264',
                         'Bitrate' => 1500,
                         'Width' => 1280,
                         'Fps' => 25);
# Ouput->Audio
$output['Audio'] = array('Codec' => 'AAC',
                         'Bitrate' => 128,
                         'Channels' => 2,
                         'Samplerate' => 44100);
# Ouput->TemplateId
$output['TemplateId'] = $template_id;
$outputs = array($output);
$request->setOUtputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# 发起请求并处理返回
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";;
    if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
```

```
        print 'JobId is:' .
              $response->{'JobResultList'}->{'JobResult'}[0]->{'Job
'}->{'JobId'} . "\n";
    } else {
        print 'SubmitJobs Failed code:' .
              $response->{'JobResultList'}->{'JobResult'}[0]->{'Code
'} .
              ' message:' .
              $response->{'JobResultList'}->{'JobResult'}[0]->{'
Message'} . "\n";
    }
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
}
```

3.5.5 水印

1. **创建AcsClient实例。**

```
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                  # Region ID
    $access_key_id,                  # AccessKey ID
    $access_key_secret               # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
```

2. **创建request，并设置参数。**

```
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
```

3. **水印参数。**

   - **图片水印**

```
$image_watermark_input = array(
                     'Location' => $oss_location,
                     'Bucket' => $oss_bucket,
                     'Object' => urlencode($image_watermark_object
)
                    );
 $image_watermark = array(
             'WaterMarkTemplateId' => $watermark_template_id,
             'Type' => 'Image',
             'InputFile' => $image_watermark_input,
             'ReferPos' => 'TopRight',
             'Width' => 0.05,
             'Dx' => 0,
             'Dy'=> 0
            );
```

   - **文字水印**

```
$text_config = array(
          'Content' => '5rWL6K+V5paH5a2X5rC05Y2w',
          'FontName' => 'SimSun',
```

```
            'FontSize' => 16,
            'FontColor' => 'Red',
            'FontAlpha' => 0.5,
            'Top' => 10,
            'Left' => 10
           );
 $text_watermark = array(
              'WaterMarkTemplateId' => $watermark_template_id,
              'Type' => 'Text',
              'TextWaterMark' => $text_config
             );
```

- **视频水印**

```
$video_watermark_input = array (
                     'Location' => $oss_location,
                     'Bucket' => $oss_bucket,
                     'Object' => urlencode($video_watermark_object)
                    );
 $video_watermark = array(
              'WaterMarkTemplateId' => $watermark_template_id,
              'Type' => 'Image',
              'InputFile'=> $video_watermark_input,
              'ReferPos' => 'BottomLeft',
              'Height' => 240,
              'Dx' => 0,
              'Dy' => 0
             );
```

4. 发起API请求并显示返回值。

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";;
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
    print 'JobId is:' .
           $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}-
>{'JobId'} . "\n";
} else {
    print 'SubmitJobs Failed code:' .
           $response->{'JobResultList'}->{'JobResult'}[0]->{'Code
'} .
           ' message:' .
           $response->{'JobResultList'}->{'JobResult'}[0]->{'Message
'} . "\n";
}
```

完整代码

```
<?php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$watermark_template_id = 'xxx';
$template_id = 'S00000001-200030';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'presigned';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
$image_watermark_object = 'logo.png';
```

```
$video_watermark_object = 'logo.mov';
# DefaultAcsClient
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                  # Region ID
    $access_key_id,                  # AccessKey ID
    $access_key_secret               # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# request
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
               'Bucket' => $oss_bucket,
               'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Ouput->TemplateId
$output['TemplateId'] = $template_id;
## Image Watermark
$image_watermark_input = array(
                          'Location' => $oss_location,
                          'Bucket' => $oss_bucket,
                          'Object' => urlencode($image_watermark_obje
ct)
                         );
$image_watermark = array(
                  'WaterMarkTemplateId' => $watermark_template_id,
                  'Type' => 'Image',
                  'InputFile' => $image_watermark_input,
                  'ReferPos' => 'TopRight',
                  'Width' => 0.05,
                  'Dx' => 0,
                  'Dy'=> 0
                 );
## Text Watermark
$text_config = array(
               'Content' => '5rWL6K+V5paH5a2X5rC05Y2w',
               'FontName' => 'SimSun',
               'FontSize' => 16,
               'FontColor' => 'Red',
               'FontAlpha' => 0.5,
               'Top' => 10,
               'Left' => 10
              );
$text_watermark = array(
                  'WaterMarkTemplateId' => $watermark_template_id,
                  'Type' => 'Text',
                  'TextWaterMark' => $text_config
                 );
## Video Watermark
$video_watermark_input = array (
                         'Location' => $oss_location,
                         'Bucket' => $oss_bucket,
                         'Object' => urlencode($video_watermark_object
)
                        );
$video_watermark = array(
                  'WaterMarkTemplateId' => $watermark_template_id,
                  'Type' => 'Image',
                  'InputFile'=> $video_watermark_input,
                  'ReferPos' => 'BottomLeft',
                  'Height' => 240,
```

```
                    'Dx' => 0,
                    'Dy' => 0
                   );
# Output->Watermarks
$watermarks = array($image_watermark, $text_watermark, $video_wate
rmark);
$output['WaterMarks'] = $watermarks;
# Outputs
$outputs = array($output);
$request->setOUtputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";;
    if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
        print 'JobId is:' .
               $response->{'JobResultList'}->{'JobResult'}[0]->{'Job
'}->{'JobId'} . "\n";
    } else {
        print 'SubmitJobs Failed code:' .
               $response->{'JobResultList'}->{'JobResult'}[0]->{'Code
'} .
               ' message:' .
               $response->{'JobResultList'}->{'JobResult'}[0]->{'
Message'} . "\n";
    }
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
}
```

3.5.6 截图

1. **创建AcsClient实例。**

```
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                  # Region ID
    $access_key_id,                  # AccessKey ID
    $access_key_secret               # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
```

2. **创建request，并设置参数。**

```
$request = new Mts\SubmitSnapshotJobRequest();
$request->setAcceptFormat('JSON');
```

3. **截图参数。**

   - **Input**

```
$input = array('Location' => $oss_location,
          'Bucket' => $oss_bucket,
```

```
            'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
```

- **SnapshotConfig**
  - **OutputFile**

```
$output = array('Location' => $oss_location,
        'Bucket' => $oss_bucket,
        'Object' => urlencode($oss_output_object));
$snapshot_config = array('OutputFile' => $output);
```

  - **Time**

```
$snapshot_config['Time'] = 2;
```

  - **Interval/Num**

```
$snapshot_config['Interval'] = 2;
$snapshot_config['Num'] = 3;
```

  - **Width/Height**

```
$snapshot_config['Height'] = 360;
```

4. 发起API请求并显示返回值。

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";
print 'JobId is:' . $response->{'SnapshotJob'}->{'Id'} . "\n";
print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/
output_00001.jpg' . "\n";
print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/
output_00002.jpg' . "\n";
print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/
output_00003.jpg' . "\n";
```

完整代码

```
<?php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output_{Count}.jpg';
# DefaultAcsClient
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                    # Region ID
    $access_key_id,                    # AccessKey ID
    $access_key_secret                 # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# request
$request = new Mts\SubmitSnapshotJobRequest();
$request->setAcceptFormat('JSON');
```

```
# Input
$input = array('Location' => $oss_location,
               'Bucket' => $oss_bucket,
               'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# SnapshotConfig->OutputFile
$output = array('Location' => $oss_location,
                'Bucket' => $oss_bucket,
                'Object' => urlencode($oss_output_object));
$snapshot_config = array('OutputFile' => $output);
# SnapshotConfig->Time
$snapshot_config['Time'] = 2;
# SnapshotConfig->Interval/Num
$snapshot_config['Interval'] = 2;
$snapshot_config['Num'] = 3;
# SnapshotConfig->Width/Height
$snapshot_config['Height'] = 360;
# SnapshotConfig
$request->setSnapshotConfig(json_encode($snapshot_config));
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";
    print 'JobId is:' . $response->{'SnapshotJob'}->{'Id'} . "\n";
    print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/
output_00001.jpg' . "\n";
    print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/
output_00002.jpg' . "\n";
    print 'http://'.$oss_bucket.'.'.$oss_location.'.aliyuncs.com/
output_00003.jpg' . "\n";
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
}
```

3.5.7 拼接和简单剪辑

1. **创建AcsClient实例。**

```
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                  # Region ID
    $access_key_id,                  # AccessKey ID
    $access_key_secret               # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
```

2. **创建request，并设置参数。**

```
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
```

3. **转码参数。**

- Input

```
$input = array('Location' => $oss_location,
          'Bucket' => $oss_bucket,
          'Object' => urlencode($head_object));
$request->setInput(json_encode($input));
```

- Output

```
$output = array('OutputObject' => urlencode($oss_output_object));
```

  - Video

```
$output['Video'] = array('Width' => 1280,
                 'Height' => 720);
```

  - MergeList

```
$merge_video = array('MergeURL' => 'http://'.$oss_bucket.'.'.$
oss_location.'.aliyuncs.com/'.urlencode($oss_input_object));
$merge_tail = array('MergeURL' => 'http://'.$oss_bucket.'.'.$
oss_location.'.aliyuncs.com/'.urlencode($tail_object));
$output['MergeList'] = array($merge_video, $merge_tail);
```

4. 发起API请求并显示返回值。

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";;
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
    print 'JobId is:' .
          $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}-
>{'JobId'} . "\n";
} else {
    print 'SubmitJobs Failed code:' .
          $response->{'JobResultList'}->{'JobResult'}[0]->{'Code
'} .
          ' message:' .
          $response->{'JobResultList'}->{'JobResult'}[0]->{'Message
'} . "\n";
}
```

完整代码

```
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$template_id = 'S00000001-200030';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
$head_object = 'head.mp4';
$tail_object = 'tail.mp4';
# 创建DefaultAcsClient实例并初始化
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                    # Region ID
```

```
    $access_key_id,                    # AccessKey ID
    $access_key_secret                 # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# 创建API请求并设置参数
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
               'Bucket' => $oss_bucket,
               'Object' => urlencode($head_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Ouput->Video
$output['Video'] = array('Width' => 1280,
                         'Height' => 720);
# Ouput->TemplateId
$output['TemplateId'] = $template_id;
# Output->MergeList
$merge_video = array('MergeURL' => 'http://'.$oss_bucket.'.'.$
oss_location.'.aliyuncs.com/'.urlencode($oss_input_object));
$merge_tail = array('MergeURL' => 'http://'.$oss_bucket.'.'.$
oss_location.'.aliyuncs.com/'.urlencode($tail_object));
$output['MergeList'] = array($merge_video, $merge_tail);
# Outputs
$outputs = array($output);
$request->setOUtputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";;
    if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
        print 'JobId is:' .
               $response->{'JobResultList'}->{'JobResult'}[0]->{'Job
'}->{'JobId'} . "\n";
    } else {
        print 'SubmitJobs Failed code:' .
               $response->{'JobResultList'}->{'JobResult'}[0]->{'Code
'} .
               ' message:' .
               $response->{'JobResultList'}->{'JobResult'}[0]->{'
Message'} . "\n";
    }
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
}
```

3.5.8 拼接-开板和尾板

1. **创建AcsClient实例**。

```
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                    # Region ID
```

```
    $access_key_id,                    # AccessKey ID
    $access_key_secret                 # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
```

2. 创建request，并设置参数。

```
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
```

3. 转码参数。

   · Input

   ```
   $input = array('Location' => $oss_location,
             'Bucket' => $oss_bucket,
             'Object' => urlencode($head_object));
   $request->setInput(json_encode($input));
   ```

   · Output

   ```
   $output = array('OutputObject' => urlencode($oss_output_object));
   ```

   - Video

     ```
     $output['Video'] = array('Width' => 1280,
                      'Height' => 720);
     ```

   - OpeningList

     ```
     $opening_video = array('OpenUrl' => 'http://'.$oss_bucket.'.'.$
     oss_location.'.aliyuncs.com/'.urlencode($head_object),
                    'Width' => 640,
                    'Start' => 2);
     $output['OpeningList'] = array($opening_video);
     ```

   - TailSlateList

     ```
     $tailslate_video = array('TailUrl' => 'http://'.$oss_bucket
     .'.'.$oss_location.'.aliyuncs.com/'.urlencode($tail_object),
                      'Width' => 640,
                      'BlendDuration' => 3,
                      'BgColor' => 'Black');
     $output['TailSlateList'] = array($tailslate_video);
     ```

4. 发起API请求并显示返回值。

```
$response = $client->getAcsResponse($request);
print 'RequestId is:' . $response->{'RequestId'} . "\n";;
if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
    print 'JobId is:' .
           $response->{'JobResultList'}->{'JobResult'}[0]->{'Job'}-
>{'JobId'} . "\n";
} else {
    print 'SubmitJobs Failed code:' .
           $response->{'JobResultList'}->{'JobResult'}[0]->{'Code
'} .
           ' message:' .
```

```
            $response->{'JobResultList'}->{'JobResult'}[0]->{'Message
'} . "\n";
}
```

完整代码

```
<?php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
$access_key_id = 'xxx';
$access_key_secret = 'xxx';
$mps_region_id = 'cn-hangzhou';
$pipeline_id = 'xxx';
$template_id = 'S00000001-200030';
$oss_location = 'oss-cn-hangzhou';
$oss_bucket = 'xxx';
$oss_input_object = 'input.mp4';
$oss_output_object = 'output.mp4';
$head_object = 'head.mp4';
$tail_object = 'tail.mp4';
# 创建DefaultAcsClient实例并初始化
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,                    # Region ID
    $access_key_id,                    # AccessKey ID
    $access_key_secret                 # AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
# 创建API请求并设置参数
$request = new Mts\SubmitJobsRequest();
$request->setAcceptFormat('JSON');
# Input
$input = array('Location' => $oss_location,
               'Bucket' => $oss_bucket,
               'Object' => urlencode($oss_input_object));
$request->setInput(json_encode($input));
# Output
$output = array('OutputObject' => urlencode($oss_output_object));
# Ouput->TemplateId
$output['TemplateId'] = $template_id;
# Output->OpeningList
$opening_video = array('OpenUrl' => 'http://'.$oss_bucket.'.'.$
oss_location.'.aliyuncs.com/'.urlencode($head_object),
                       'Width' => 640,
                       'Start' => 2);
$output['OpeningList'] = array($opening_video);
# Output->TailSlateList
$tailslate_video = array('TailUrl' => 'http://'.$oss_bucket.'.'.$
oss_location.'.aliyuncs.com/'.urlencode($tail_object),
                         'Width' => 640,
                         'BlendDuration' => 3,
                         'BgColor' => 'Black');
$output['TailSlateList'] = array($tailslate_video);
# Outputs
$outputs = array($output);
$request->setOUtputs(json_encode($outputs));
$request->setOutputBucket($oss_bucket);
$request->setOutputLocation($oss_location);
# PipelineId
$request->setPipelineId($pipeline_id);
# call api
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";;
```

```
    if ($response->{'JobResultList'}->{'JobResult'}[0]->{'Success'}) {
        print 'JobId is:' .
              $response->{'JobResultList'}->{'JobResult'}[0]->{'Job
'}->{'JobId'} . "\n";
    } else {
        print 'SubmitJobs Failed code:' .
              $response->{'JobResultList'}->{'JobResult'}[0]->{'Code
'} .
              ' message:' .
              $response->{'JobResultList'}->{'JobResult'}[0]->{'
Message'} . "\n";
    }
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
}
```

3.5.9 创建HLS标准加密工作流

简介

示例调用API进行创建HLS标准加密工作流。创建工作流，参见 新增媒体工作流。MPS SDK，参见 安装。

示例代码

```
<?php
include_once 'aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
date_default_timezone_set('PRC');
class HLSEncryptionWorkflowDemo {
    private $client;
    private $region = '<region>';
    private $accessKeyId = '<accessKeyId>';
    private $accessKeySecret = '<accessKeySecret>';
    private $pipelineId = "<PipelineId>";
    private $templateId = "S00000001-100020"; #转码模版ID，m3u8模版，按需
配置
    private $ossLocation = "<OssLocation>";
    private $inputBucket = "<InputBucket>";
    private $inputPath = "<InputPath>"; #如 "HLS-Encryption"
    private $outputBucket = "<OutputBucket>";
    private $encryptionType = "hls-aes-128";
    private $hlsKeyUri = "<解密密钥的URI>"; #如http://decrypt.testdomain
.com
    private $actStart = "Act-Start";
    private $actEncryption = "Act-HLS-Encryption";
    private $actReport = "Act-Report";
    function __construct() {
        $profile = DefaultProfile::getProfile($this->region, $this->
accessKeyId, $this->accessKeySecret);
        $this->client = new DefaultAcsClient($profile);
    }
    function addMediaWorkflow() {
        $request = new Mts\AddMediaWorkflowRequest();
        $request->setName("HLS加密工作流php");
        $request->setTopology($this->buildWorkflowTopology());
```

```
        $response = $this->client->getAcsResponse($request);
        echo json_encode($response);
    }
    function buildWorkflowTopology() {
        $activities = $this->buildActivities();
        $dependencies = $this->buildDependencies();
        $workflow = array(
                          "Activities" => $activities,
                          "Dependencies" => $dependencies
                         );
        echo json_encode($workflow)."\n";
        return json_encode($workflow);
    }
    function buildActivities() {
        $activities = [
            $this->actStart => $this->buildStartActivity(),
            $this->actEncryption => $this->buildTranscodeActivity(),
            $this->actReport => $this->buildReportActivity()
        ];
        return $activities;
    }
    function buildStartActivity() {
        $startActivity = array(
            "Name" => $this->actStart,
            "Type" => "Start",
            "Parameters" => $this->buildStartParameters()
        );
        return $startActivity;
    }
    function buildStartParameters() {
        $startParameters = array(
            "PipelineId" => $this->pipelineId,
            "InputFile" => $this->buildInputFile()
        );
        return $startParameters;
    }
    function buildInputFile() {
        $inputFile = array(
            "Bucket" => $this->inputBucket,
            "Location" => $this->ossLocation,
            "ObjectPrefix" => $this->inputPath
        );
        return $inputFile;
    }
    function buildTranscodeActivity() {
        $transcodeParameters = array(
            "Name" => $this->actEncryption,
            "Type" => "Transcode",
            "Parameters" => $this->buildTranscodeParameters()
        );
        return $transcodeParameters;
    }
    function buildTranscodeParameters() {
        $transcodeParameters = array(
            "OutputBucket" => $this->outputBucket,
            "OutputLocation" => $this->ossLocation,
            "Outputs" => $this->buildOutputsConfig()
        );
        return $transcodeParameters;
    }
    function buildOutputsConfig() {
        $output = array(
            "ObjectRegex" => $this->actEncryption."/{RunId}/{FileName
}",
```

```
            "TemplateId" => $this->templateId,
            "Encryption" => $this->buildEncryption()
        );
        $outputs = array($output);
        return $outputs;
    }
    function buildEncryption() {
        $encryption = array(
            "Type" => $this->encryptionType,
            "KeyUri" => $this->hlsKeyUri
        );
        return $encryption;
    }
    function buildReportActivity() {
        $reportActivity = array(
            "Name" => $this->actReport,
            "Parameters" => (object)[],
            "Type" => "Report"
        );
        return $reportActivity;
    }
    function buildDependencies() {
        $subActivityOfStart = array(
            $this->actEncryption
        );
        $subActivityOfTranscode = array(
            $this->actReport
        );
        $dependencies = array(
            $this->actStart => $subActivityOfStart,
            $this->actEncryption => $subActivityOfTranscode,
            $this->actReport => []
        );
        return $dependencies;
    }
}
$demo = new HLSEncryptionWorkflowDemo();
$demo->addMediaWorkflow();
?>
```

3.5.10 管道管理

在开通服务时，系统会自动创建一个用户管道。您还可以通过一系列接口管理管道（pipeline）。

例如：SearchPipeline、QueryPipelineList、UpdatePipeline。

搜索管道

可以直接通过SearchPipeline**接口搜索管道信息。**

```
$region = '<region>';
    $accessKeyId = '<accessKeyId>';
    $accessKeySecret = '<accessKeySecret>';
    $profile = DefaultProfile::getProfile($region, $accessKeyId, $
accessKeySecret);
    $client = new DefaultAcsClient($profile);
    $request = new Mts\SearchPipelineRequest();
    // 如果出错，可能会抛出ClientException或ServerException异常
    $response = $client->getAcsResponse($request);
    $pipelines = $response->PipelineList->Pipeline;
    foreach ($pipelines as $pipeline) {
```

```
        echo 'pipeline id:' . $pipeline->Id . ', name:' . $pipeline->
Name . ', state:' . $pipeline->State . "\n";
    }
```

查询管道

如果已经知道pipelineId，可以通过pipelineId调用QueryPipelineList接口查询管道信息。

```
$region = '<region>';
    $accessKeyId = '<accessKeyId>';
    $accessKeySecret = '<accessKeySecret>';
    // 已知的管道ID，多个管道用','分隔
    $pipelineIds = '<pipelineIds>';
    $profile = DefaultProfile::getProfile($region, $accessKeyId, $
accessKeySecret);
    $client = new DefaultAcsClient($profile);
    $request = new Mts\QueryPipelineListRequest();
    $request->setPipelineIds($pipelineIds);
    // 如果出错，可能会抛出ClientException或ServerException异常
    $response = $client->getAcsResponse($request);
    $pipelines = $response->PipelineList->Pipeline;
    foreach ($pipelines as $pipeline) {
        echo 'pipeline id:' . $pipeline->Id . ', name:' . $pipeline->
Name . ', state:' . $pipeline->State . "\n";
    }
```

更新管道

通过UpdatePipeline更新管道信息，包括更新管道名称，状态。管道的状态包括Active、Paused。

```
$region = '<region>';
    $accessKeyId = '<accessKeyId>';
    $accessKeySecret = '<accessKeySecret>';
    $profile = DefaultProfile::getProfile($region, $accessKeyId, $
accessKeySecret);
    $client = new DefaultAcsClient($profile);
    $request = new Mts\SearchPipelineRequest();
    // 如果出错，可能会抛出ClientException或ServerException异常
    $response = $client->getAcsResponse($request);
    $pipelines = $response->PipelineList->Pipeline;
    $pipeline = $pipelines[0];
    $request = new Mts\UpdatePipelineRequest();
    $request->setPipelineId($pipeline->Id);
    $request->setName($pipeline->Name);
    $request->setState($pipeline->State == 'Paused' ? 'Active' : '
Paused');
    $response = $client->getAcsResponse($request);
    $pipeline = $response->Pipeline;
    echo 'pipeline id:' . $pipeline->Id . ', name:' . $pipeline->Name
 . ', state:' . $pipeline->State . "\n";
```

3.5.11 查询媒体-使用OSS文件地址

如果您不知道媒体ID（直播走工作流转点播），可以通过媒体的输入地址进行媒体信息的查询，接口为 QueryMediaListByURL。

```
<?php
```

```
include_once 'aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
date_default_timezone_set('PRC');
class QueryMediaListByURLDemo {
    private $client;
    private $region = '<region>';
    private $accessKeyId = '<accessKeyId>';
    private $accessKeySecret = '<accessKeySecret>';
    function __construct()
    {
        $profile = DefaultProfile::getProfile($this->region, $this->
accessKeyId, $this->accessKeySecret);
        $this->client = new DefaultAcsClient($profile);
    }
    function queryMediaListByUrl()
    {
        $request = new Mts\QueryMediaListByURLRequest();
        $ossDomain = 'http://<input-bucket>.<region>.aliyuncs.com/';
        #ossObject需要RFC3986编码
        $ossObject = $this->encodeByRFC3986('test/笑傲江湖.mp4');
        $request->setFileURLs($ossDomain.$ossObject);
        $response = $this->client->getAcsResponse($request);
        echo json_encode($response);
    }
    function encodeByRFC3986($arg_1)
    {
        $encodeOssObject="";
        $arraylist = explode("/", $arg_1);
        for($i = 0; $i < count($arraylist); $i++)
        {
            $tmp = rawurlencode($arraylist[$i]);
            $encodeOssObject = $encodeOssObject.$tmp;
            if ($i !== count($arraylist) -1) {
                $encodeOssObject = $encodeOssObject."/";
            }
        }
        return $encodeOssObject;
    }
}
$demo = new QueryMediaListByURLDemo();
$demo->queryMediaListByUrl();
?>
```

3.5.12 新增媒体

新增媒体，添加视频文件到媒体库。

```
<?php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
# Step 1 set region
$REGION = "cn-shenzhen";
$OSS_REGION = "oss-cn-shenzhen";
$mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
$accessKeyId = "";
$accessKeySecret = "";
# 创建DefaultAcsClient实例并初始化
$clientProfile = DefaultProfile::getProfile(
    $REGION,                          # 您的 Region ID
    $accessKeyId,                     # 您的 AccessKey ID
    $accessKeySecret                  # 您的 AccessKey Secret
```

```
);
$client = new DefaultAcsClient($clientProfile);
$request = new Mts\AddMediaRequest();
$request->setAcceptFormat('JSON');
$request->setFileURL("http://mtb-sz-in.oss-cn-shenzhen.aliyuncs.com/
media/r180-ABC.mp4");
$request->setMediaWorkflowId("829bed0300994057a49e4f16de957e34");
# 发起请求并处理返回
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";;
    print "Response:".json_encode($response);
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
}
?>
```

3.5.13 新增媒体工作流

新增媒体工作流，您可以对MPS服务提供的活动（例如，转码、截图等活动）进行组装，拓扑结构如下。

```
<?php
include_once 'aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php';
use Mts\Request\V20140618 as Mts;
# Step 1 set region
$REGION = "cn-shenzhen";
$OSS_REGION = "oss-cn-shenzhen";
$mtsEndpoint = "mts." + REGION + ".aliyuncs.com";
# Step 2.set accesskey & keySecret
$accessKeyId = "";
$accessKeySecret = "";
# Step 3.set mps transcoding queue id
$PIPELINE_ID = "38bba54d524448be92d277caaa8da118";
# 创建DefaultAcsClient实例并初始化
$clientProfile = DefaultProfile::getProfile(
    $REGION,                          # 您的 Region ID
    $accessKeyId,                     # 您的 AccessKey ID
    $accessKeySecret                  # 您的 AccessKey Secret
);
$client = new DefaultAcsClient($clientProfile);
$request = new Mts\AddMediaWorkflowRequest();
$request->setAcceptFormat('JSON');
$request->setName("Sequential-workflow");
$startActivity = array(
        "Type"=>"Start",
        "Parameters"=>array(
            "InputFile"=>array(
                "Bucket"=> "mtb-sz-in",
                "Location"=> $OSS_REGION,
                "ObjectPrefix"=> "media/"
            ),
            "PipelineId"=>$PIPELINE_ID
        )
    );
$transcodeActivity = array(
        "Type"=>"Transcode",
        "Parameters"=> array (
```

```
            "Outputs"=>array(
                array(
                    "OutputObject"=> urlencode("transcode/{ObjectPref
ix}/{FileName}.{ExtName}"),
                    "TemplateId"=> "S00000001-000070"
                )
            ),
            "OutputLocation"=> $OSS_REGION,
            "OutputBucket"=>"mtb-sz-out"
        )
    );
$reportActivity = array(
        "Type"=> "Report",
        "Parameters"=> array(
             "PublishType"=>"Auto"
        )
    );
$topology = array(
        "Activities"=> array(
            "startNode"=>$startActivity,
            "transcodingNode"=>$transcodeActivity,
            "reportNode"=>$reportActivity
        ),
        "Dependencies"=>array (
            "startNode"=>array("transcodingNode"),
            "transcodingNode"=>array("reportNode"),
            "reportNode"=>array()
        )
    );
$request->setTopology(json_encode($topology));
# 发起请求并处理返回
try {
    $response = $client->getAcsResponse($request);
    print 'RequestId is:' . $response->{'RequestId'} . "\n";;
    print "Response:".json_encode($response);
} catch(ServerException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
} catch(ClientException $e) {
    print 'Error: ' . $e->getErrorCode() . ' Message: ' . $e->
getMessage() . "\n";
}
?>
```

4 Demo 工程

目前阿里云官方提供的媒体处理服务 SDK 分为 Java、Python和PHP三种语言版本，具体安装和使用详情说明如下。