

ALIBABA CLOUD

# 阿里云

媒体处理  
SDK参考

文档版本：20210112

 阿里云

## 法律声明

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

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

# 通用约定

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

# 目录

- 1.创建AccessKey ----- 07
- 2.上传SDK ----- 08
- 3.媒体转码SDK ----- 09
  - 3.1. SDK 文档简介 ----- 09
  - 3.2. 多地域支持 ----- 09
  - 3.3. Java SDK ----- 09
    - 3.3.1. 前言 ----- 09
    - 3.3.2. 安装 ----- 09
    - 3.3.3. 快速入门 ----- 12
    - 3.3.4. 转码 ----- 14
    - 3.3.5. 水印 ----- 18
    - 3.3.6. 截图 ----- 23
    - 3.3.7. HLS数据加密 ----- 27
    - 3.3.8. 创建HLS标准加密工作流 ----- 38
    - 3.3.9. 管道管理 ----- 41
    - 3.3.10. 查询媒体-使用OSS文件地址 ----- 43
    - 3.3.11. 新增媒体 ----- 45
    - 3.3.12. 视频拼接和简单剪辑 ----- 46
    - 3.3.13. 新增媒体工作流 ----- 52
    - 3.3.14. 拼接-开板和尾板 ----- 57
  - 3.4. Python SDK ----- 62
    - 3.4.1. 前言 ----- 62
    - 3.4.2. 安装 ----- 62
    - 3.4.3. 快速入门 ----- 63
    - 3.4.4. 转码 ----- 64
    - 3.4.5. 水印 ----- 67

---

3.4.6. 截图	71
3.4.7. 查询媒体-使用OSS文件地址	74
3.4.8. 创建HLS标准加密 workflow	74
3.4.9. 拼接和简单剪辑	77
3.4.10. 拼接-开板和尾板	79
3.4.11. 添加媒体	82
3.4.12. 管道管理	83
3.4.13. 添加媒体 workflow	85
3.5. PHP SDK	87
3.5.1. 前言	87
3.5.2. 安装	87
3.5.3. 快速入门	88
3.5.4. 转码	89
3.5.5. 水印	93
3.5.6. 截图	98
3.5.7. 拼接和简单剪辑	100
3.5.8. 拼接-开板和尾板	103
3.5.9. 创建HLS标准加密 workflow	106
3.5.10. 管道管理	110
3.5.11. 查询媒体-使用OSS文件地址	111
3.5.12. 新增媒体	113
3.5.13. 新增媒体 workflow	114
4.智能标签SDK	116
4.1. Java SDK	116
4.2. Python SDK	126
4.3. Php SDK	133
5.播放器SDK	144
6.Demo 工程	145

---



# 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



## 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 Git Hub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4.5.3版本SDK的核心库和2.7.6版本的媒体处理SDK为例：

```

<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.5.3</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-mts</artifactId>
  <version>2.7.6</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>4.5.3</version>
    </dependency>
    <dependency>
      <groupId>com.aliyun</groupId>
      <artifactId>aliyun-java-sdk-mts</artifactId>
      <version>2.7.6</version>
    </dependency>
    <dependency>
      <groupId>com.alibaba</groupId>
      <artifactId>fastjson</artifactId>
      <version>1.2.46</version>
    </dependency>
  </dependencies>
  <build>
    <finalName>${artifactId}-${version}</finalName>
  </build>
</project>
```

```
<plugins>
  <plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.3.2</version>
    <configuration>
      <source>1.8</source>
      <target>1.8</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AcscClient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。

- o 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());
```

- o 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.getAcResponse(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

```
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);
    // Output->Container
    JSONObject container = new JSONObject();
    container.put("Format", "mp4");
    output.put("Container", container.toJSONString());
    // Output->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.getJobResultList().get(0).getJob().getJobId());
    } else {
        System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0).getCode() +
            " message:" + response.getJobResultList().get(0).getMessage());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

## 3.3.5. 水印

1. 创建AcscClient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。

- o 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("imageWatermark 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");
```

- o 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("videoWatermark 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.getAcResponse(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);
// Output->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("imageWatermark 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("videoWatermark 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.setOutputs(outputs.toJSONString());
request.setOutputBucket(ossBucket);
request.setOutputLocation(ossLocation);
// PipelineId
request.setPipelineId(pipelineId);
// call api
SubmitJobsResponse response;
try {
    response = client.getAcResponse(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());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

### 3.3.6. 截图

1. 创建AcClient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。

- o 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());
```

- o Snapshot Config

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

- Output File

```
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.getAcResponse(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 SubmitSnapshotJobRequest();
// 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。
- 如果对安全有强需求，请开启 workflow 中的数据加密。详情参见 [数据加密](#)。

#### 使用限制

- HLS标准数据加密需要使用SubmitJobs接口。
- workflow 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>
```

## 加密示例代码（Base64方式）

```
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.alibaba.mediaprocess.mpsDocTestShanghai.Config;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
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;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

public class DataEncryptionBasicBase64 {
    private static String accessKeyId = "****"; // RAM账号的AccessKey ID
    private static String accessKeySecret = "****"; // RAM账号Access Key Secret
    private static String mpsRegionId="cn-shanghai"; // 地域ID
    private DefaultAcsClient client = null;
    private final String location = "oss-cn-shanghai"; // 按需配置
    private final String ossInputBucket = "huadong2-test"; // 按需配置
    private final String ossOutputBucket = "huadong2output"; // 按需配置
    private final String pipelineId = "f24c8a87a497456898b467892eea84d6"; // 按需配置
    public DataEncryptionBasicBase64() throws ClientException {
        DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId, // 地域ID
            accessKeyId, // RAM账号的AccessKey ID
            accessKeySecret); // RAM账号Access Key Secret
        this.client = new DefaultAcsClient(profile);
    }
    private JSONObject getInputFile() {
```

```
JSONObject inputFile = new JSONObject();
inputFile.put("Location", location);
inputFile.put("Bucket", ossInputBucket);
try {
    inputFile.put("Object", URLEncoder.encode("wb-ceshi/487_93.mp4", "utf-8"));
} catch (UnsupportedEncodingException e) {
    throw new RuntimeException("input URL encode failed");
}
return inputFile;
}

private JSONArray getOutputs() throws ClientException{
    JSONArray outputs = new JSONArray();
    outputs.add(getOutput());
    return outputs;
}

private JSONObject getOutput() throws ClientException{
    JSONObject output = new JSONObject();
    output.put("OutputObject", "wb-ceshi/202011131343");
    output.put("TemplateId", "S00000001-100000");
    output.put("Encryption", getEncryptionConfigs());
    return output;
}

private JSONObject getEncryptionConfigs() throws ClientException {
    JSONObject encryption = new JSONObject();
    encryption.put("Type", "hls-aes-128");
    // 加密字符串必须是16位
    encryption.put("Key", Base64.encodeBase64URLSafeString("encryptionkey128".getBytes()));
    String url="http://127.0.0.1:8888/";
    encryption.put("KeyUri", Base64.encodeBase64URLSafeString(url.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(ossOutputBucket);
    request.setOutputs(getOutputs().toJSONString());
    request.setPipelineId(pipelineId);
    SubmitJobsResponse response = this.client.getResponse(request);
}
```

```
        submitJobsResponse reponse = this.client.getAcsResponse(request);
        System.out.println(JSONObject.toJSONString(reponse.getJobResultList()));
        return reponse.getJobResultList().get(0).getJob().getJobId();
    }

    public static void main(String[] args) throws ClientException {
        DataEncryptionBasicBase64 demo = new DataEncryptionBasicBase64();
        String jobId = demo.submitJobs();
        System.out.println(jobId);
    }
}
```

## 加密示例代码（KMS方式）

```
import com.alibaba.fastjson.JSONArray;
import com.alibaba.fastjson.JSONObject;
import com.alibaba.aliyuncs.mpsDocTestShanghai.Config;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.kms.model.v20160120.GenerateDataKeyRequest;
import com.aliyuncs.kms.model.v20160120.GenerateDataKeyResponse;
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;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;

public class DataEncryptionBasicKMS {
    private static String accessKeyId = "****"; // RAM账号的AccessKey ID
    private static String accessKeySecret = "****"; // RAM账号Access Key Secret
    private static String mpsRegionId = "cn-shanghai"; // 地域ID
    private DefaultAcsClient client = null;
    private final String location = "oss-cn-shanghai"; // 按需配置
    private final String ossInputBucket = "huadong2-test"; // 按需配置
    private final String ossOutputBucket = "huadong2output"; // 按需配置
    private final String pipelineId = "f24c8a87a497456898b467892eea84d6"; // 按需配置
    public DataEncryptionBasicKMS() throws ClientException {
        DefaultProfile profile = DefaultProfile.getProfile(
            mpsRegionId, // 地域ID
            accessKeyId, // RAM账号的AccessKey ID
            accessKeySecret); // RAM账号Access Key Secret
        this.client = new DefaultAcsClient(profile);
    }
}
```

```
}
private JSONObject getInputFile() {
    JSONObject inputFile = new JSONObject();
    inputFile.put("Location", location);
    inputFile.put("Bucket", ossInputBucket);
    try {
        inputFile.put("Object", URLEncoder.encode("wb-ceshi/487_93.mp4", "utf-8"));
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("input URL encode failed");
    }
    return inputFile;
}
private JSONArray getOutputs() throws ClientException{
    JSONArray outputs = new JSONArray();
    outputs.add(getOutput());
    return outputs;
}
private JSONObject getOutput() throws ClientException{
    JSONObject output = new JSONObject();
    output.put("OutputObject", "wb-ceshi/20201117");
    output.put("TemplateId", "S00000001-100020");
    output.put("Encryption", getEncryptionConfigs());
    return output;
}
private JSONObject getEncryptionConfigs() throws ClientException {
    GenerateDataKeyRequest generateDataKeyRequest=new GenerateDataKeyRequest();
    generateDataKeyRequest.setKeyId("504d60ef-93f3-48e8-9d94-075aeb7d7003");
    generateDataKeyRequest.setKeySpec("AES_128");
    GenerateDataKeyResponse generateDataKeyResponse=client.getAcResponse(generateDataKeyRequest);
    JSONObject encryption = new JSONObject();
    encryption.put("Type", "hls-aes-128");
    encryption.put("Key", generateDataKeyResponse.getCiphertextBlob());
    String url="http://127.0.0.1:8888/?Ciphertext="+generateDataKeyResponse.getCiphertextBlob();
    encryption.put("KeyUri", Base64.encodeBase64URLSafeString(url.getBytes()));
    encryption.put("KeyType", "KMS");
    return encryption;
}
private String submitJobs() throws ClientException {
    JSONObject inputFile = getInputFile();
    SubmitJobsRequest request = new SubmitJobsRequest();
```

```
request.setInput(inputFile.toJSONString());
request.setOutputLocation(location);
request.setOutputBucket(ossOutputBucket);
request.setOutputs(getOutputs().toJSONString());
request.setPipelineId(pipelineId);
SubmitJobsResponse reponse = this.client.getAcResponse(request);
System.out.println(JSONObject.toJSONString(reponse.getJobResultList()));
return reponse.getJobResultList().get(0).getJob().getJobId();
}
public static void main(String[] args) throws ClientException {
    DataEncryptionBasicKMS demo = new DataEncryptionBasicKMS();
    String jobId = demo.submitJobs();
    System.out.println(jobId);
}
}
```

## 解密示例代码（Base64方式）

```
import com.sun.net.httpserver.Headers;
import com.sun.net.httpserver.HttpExchange;
import com.sun.net.httpserver.HttpHandler;
import com.sun.net.httpserver.HttpServer;
import com.sun.net.httpserver.spi.HttpServiceProvider;
import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.InetSocketAddress;
public class Base64DecryptServer {
    /**
     * 说明:
     * 1、接收解密请求
     * 2、将明文秘钥base64decode返回
     */
    public class Base64DecryptHandler implements HttpHandler {
        /**
         * 处理解密请求
         *
         * @param httpExchange
         * @throws IOException
         */
    }
}
```



```
public void handle(HttpExchange httpExchange) throws IOException {
    System.out.println("request in");
    String requestMethod = httpExchange.getRequestMethod();
    if ("GET".equalsIgnoreCase(requestMethod)) {
        // 此处的解密密钥需要和加密时候的密钥一致
        byte[] key = "encryptionkey128".getBytes();
        // 设置header
        setHeader(httpExchange, key);
        // 返回base64decode之后的密钥
        OutputStream responseBody = httpExchange.getResponseBody();
        System.out.println(new String(key));
        responseBody.write(key);
        responseBody.close();
    }
}

private void setHeader(HttpExchange httpExchange, byte[] key) throws IOException {
    Headers responseHeaders = httpExchange.getResponseHeaders();
    responseHeaders.set("Access-Control-Allow-Origin", "");
    httpExchange.sendResponseHeaders(URLConnection.HTTP_OK, key.length);
}

/**
 * 服务启动
 *
 * @throws IOException
 */
private void serviceBootstrap() throws IOException {
    HttpServerProvider provider = HttpServerProvider.provider();
    //监听端口9999,能同时接受30个请求
    HttpServer httpserver = provider.createHttpServer(new InetSocketAddress(8888), 30);
    httpserver.createContext("/", new Base64DecryptHandler());
    httpserver.start();
    System.out.println("hls decrypt server started");
}

public static void main(String[] args) throws IOException {
    Base64DecryptServer server = new Base64DecryptServer();
    server.serviceBootstrap();
}
}
```

## 解密示例代码（KMS方式）

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.kms.model.v20160120.DecryptRequest;
import com.aliyuncs.kms.model.v20160120.DecryptResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.sun.net.httpserver.Headers;
import com.sun.net.httpserver.HttpExchange;
import com.sun.net.httpserver.HttpHandler;
import com.sun.net.httpserver.HttpServer;
import com.sun.net.httpserver.spi.HttpServerProvider;
import org.apache.commons.codec.binary.Base64;
import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.InetSocketAddress;
import java.net.URI;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class HlsDecryptServer {
    private static DefaultAcsClient client;
    static {
        String accessKeyId = "****"; // RAM账号的AccessKey ID
        String accessKeySecret = "****"; // RAM账号Access Key Secret    String region = "cn-shanghai"; // KMS的区域，必须与视频区域对应
        client = new DefaultAcsClient(DefaultProfile.getProfile(region, accessKeyId, accessKeySecret));
    }
    /**
     * 说明：
     * 1、接收解密请求，获取密文密钥和令牌Token
     * 2、调用KMS decrypt接口获取明文密钥
     * 3、将明文密钥base64decode返回
     */
    public class HlsDecryptHandler implements HttpHandler {
        /**
         * 处理解密请求
         *
         * @param httpExchange
         * @throws IOException
         */
    }
}
```

```
*/
public void handle(HttpExchange httpExchange) throws IOException {
    System.out.println("request in");
    String requestMethod = httpExchange.getRequestMethod();
    if ("GET".equalsIgnoreCase(requestMethod)) {
        // 校验token的有效性
        String token = getMtsHlsUriToken(httpExchange);
        boolean validRe = validateToken(token);
        if (!validRe) {
            return;
        }
        // 从URL中取得密文密钥
        String ciphertext = getCiphertext(httpExchange);
        if (null == ciphertext)
            return;
        // 从KMS中解密出来, 并Base64 decode
        byte[] key = decrypt(ciphertext);
        //设置header
        setHeader(httpExchange, key);
        //返回base64decode之后的密钥
        OutputStream responseBody = httpExchange.getResponseBody();
        System.out.println(new String(key));
        responseBody.write(key);
        responseBody.close();
    }
}

private void setHeader(HttpExchange httpExchange, byte[] key) throws IOException {
    Headers responseHeaders = httpExchange.getResponseHeaders();
    responseHeaders.set("Access-Control-Allow-Origin", "");
    httpExchange.sendResponseHeaders(URLConnection.HTTP_OK, key.length);
}

/**
 * 调用KMS decrypt接口解密, 并将明文base64decode
 *
 * @param ciphertext
 * @return
 */
private byte[] decrypt(String ciphertext) {
    DecryptRequest request = new DecryptRequest();
    request.setCiphertextBlob(ciphertext);
    request.setProtocol(ProtocolType.HTTPS);
}
```

```
request.setRequestBody(requestType);
try {
    DecryptResponse response = client.getAcsResponse(request);
    String plaintext = response.getPlaintext();
    //注意：需要base64 decode
    return Base64.decodeBase64(plaintext);
} catch (ClientException e) {
    e.printStackTrace();
    return null;
}
}
/**
 * 校验令牌有效性
 *
 * @param token
 * @return
 */
private boolean validateToken(String token) {
    if (null == token || "".equals(token)) {
        return false;
    }
    //TODO 业务方实现令牌有效性校验
    return true;
}
/**
 * 从URL中获取密文密钥参数
 *
 * @param httpExchange
 * @return
 */
private String getCiphertext(HttpExchange httpExchange) {
    URI uri = httpExchange.getRequestURI();
    String queryString = uri.getQuery();
    String pattern = "Ciphertext=(\\w*)";
    Pattern r = Pattern.compile(pattern);
    Matcher m = r.matcher(queryString);
    if (m.find())
        return m.group(1);
    else {
        System.out.println("Not Found Ciphertext Param");
        return null;
    }
}
```

```
    }
}
/**
 * 获取Token参数
 *
 * @param httpExchange
 * @return
 */
private String getMtsHlsUriToken(HttpExchange httpExchange) {
    URI uri = httpExchange.getRequestURI();
    String queryString = uri.getQuery();
    String pattern = "MtsHlsUriToken=(\\w*)";
    Pattern r = Pattern.compile(pattern);
    Matcher m = r.matcher(queryString);
    if (m.find())
        return m.group(1);
    else {
        System.out.println("Not Found MtsHlsUriToken Param");
        return null;
    }
}
}
/**
 * 服务启动
 *
 * @throws IOException
 */
private void serviceBootStrap() throws IOException {
    HttpServerProvider provider = HttpServerProvider.provider();
    // 监听端口8888,能同时接受30个请求
    HttpServer httpserver = provider.createHttpServer(new InetSocketAddress(8888), 30);
    httpserver.createContext("/", new HlsDecryptHandler());
    httpserver.start();
    System.out.println("hls decrypt server started");
}
public static void main(String[] args) throws IOException {
    HlsDecryptServer server = new HlsDecryptServer();
    server.serviceBootStrap();
}
}
```

## 3.3.8. 创建HLS标准加密 workflow

### 简介

示例调用API进行创建HLS标准加密 workflow。创建工作流，参见 [新增媒体 workflow](#)。

### 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 workflow");
    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 SearchPipelineRequest();
// 如果出错，可能会抛出ClientException或ServerException异常
SearchPipelineResponse searchPipelineResponse = client.getAcsResponse(searchPipelineRequest);
List<SearchPipelineResponse.Pipeline> pipelines = searchPipelineResponse.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 QueryMediaListByURLRequest();
    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.getAcsResponse(request);
    System.out.println(JSONObject.toJSONString(response.getMediaList()));
}
private String encodeByRFC3986(String object) throws UnsupportedEncodingException {
    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 workflowID 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("829bed0300994057a49e4f16de957e34");
        try {
            AddMediaResponse 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());
    }
}
}
```

### 3.3.12. 视频拼接和简单剪辑

1. 创建AcscClient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。

- o 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());
```

- o 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(ossInputObject, "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.getAcResponse(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);
// Output->TemplateId
output.put("TemplateId", templateId);
// Output->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(ossInputObject, "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.getJobResultList().get(0).getJob().getJobId());
    } else {
        System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0).getCode() +
            " message:" + response.getJobResultList().get(0).getMessage());
    }
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

```
    },  
    }  
}
```

### 3.3.13. 新增媒体 workflow

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, Generate
    MasterPlayList, 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.DefaultAcscClient;
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 = "38bba54d524448be92d277caaa8da118";
    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。

- o 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());
```

- o 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.getJobResultList().get(0).getJob().getJobId());
    } else {
        System.out.println("SubmitJobs Failed code:" + response.getJobResultList().get(0).getCode() +
            " message:" + response.getJobResultList().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 Git Hub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

```
```\n\npip install aliyun-python-sdk-core\npip install aliyun-python-sdk-mts\n```\n
```

- Python 3.x

```
```\n\npip install aliyun-python-sdk-core-v3\npip install aliyun-python-sdk-mts\n```\n
```

### 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()\nrequest.set_accept_format('json')
```

3. 发起API请求并显示返回值。

```
response_str = client.do_action_with_exception(request)\nresponse = json.loads(response_str)\nprint 'PipelineName is:', response['PipelineList']['Pipeline'][0]['Name']\nprint '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。

- o Input

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

- o 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 aliyunsdkkms.request.v20140618 import SubmitJobRequest
```

```
from aliyunsoxkmts.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。

- o 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
}
```

- o 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。

- o Input

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

- o Snapshot Config

- Output File

```
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.aliyuncs.com/output_00001.jpg' % (oss_bucket, oss_location)
print 'http://%.%.s.aliyuncs.com/output_00002.jpg' % (oss_bucket, oss_location)
print 'http://%.%.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 aliynsdkmmts.request.v20140618 import QueryMediaListByUrlRequest
from aliynsdkcore 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 workflow

#### 简介

示例调用API，创建HLS标准加密 workflow。创建HLS标准加密 workflow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 aliynsdkmmts.request.v20140618 import AddMediaWorkflowRequest
from aliynsdkcore import client
REGION_ID = '<region>'
ACCESS_KEY_ID = '<accessKeyId>'
ACCESS_KEY_SECRET = '<accessKeySecret>'
```

```
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。

- o Input

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

- o 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。

- o Input

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

- o 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 aliyunskmts.request.v20140618 import SearchPipelineRequest
from aliyunskcore 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 aliyunskmts.request.v20140618 import QueryPipelineListRequest
from aliyunskcore 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"]
        }
    }
```

```
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。本文为您汇总了阿里云PHP SDK快速开始和相关代码。

- [阿里云PHP SDK快速开始](#)。
- [阿里云PHP SDK Git Hub](#)。

了解阿里云PHP SDK后，您可以进行媒体处理PHP SDK的安装了。详情请参见[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
```

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::addAutoloadPath("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。

- o Input

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

- o 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;
```

- o 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));
# Output->Container
$output['Container'] = array('Format' => 'mp4');
# Output->Video
$output['Video'] = array('Codec' => 'H.264',
    'Bitrate' => 1500,
    'Width' => 1280,
    'Fps' => 25);
# Output->Audio
$output['Audio'] = array('Codec' => 'AAC',
    'Bitrate' => 128,
    'Channels' => 2,
    'Samplerate' => 44100);
# Output->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AcscClient实例。

```
$clientProfile = DefaultProfile::getProfile(
    $mps_region_id,      # Region ID
    $access_key_id,     # AccessKey ID
    $access_key_secret  # AccessKey Secret
);
$client = new DefaultAcscClient($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';
$mpps_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(
    $mpps_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_object)
);
$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_watermark);
    $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AcscClient实例。

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

2. 创建request，并设置参数。

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

3. 截图参数。

- o Input

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

- o Snapshot Config

- Output File

```
$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。

- o Input

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

o 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';
$mpps_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));
# Output->Video
$output['Video'] = array('Width' => 1280,
    'Height' => 720);
# Output->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。

- o Input

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

- o 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));
# Output->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 workflow

### 简介

示例调用API进行创建HLS标准加密 workflow。创建工作流，参见 [新增媒体 workflow](#)。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 $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 HLEncryptionWorkflowDemo();
$demo->addMediaWorkflow();
```

```
queryMediaWorkflow(),  
?>
```

## 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";
$mmtsEndpoint = "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 workflow

新增媒体 workflow，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/{ObjectPrefix}/{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. 智能标签SDK

### 4.1. Java SDK

mts的Java SDK版本需要在2.7.6版本及以上。如果使用maven管理Java依赖包，您可以在工程pom.xml文件中添加下面依赖。

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.5.3</version>
</dependency>
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-mts</artifactId>
  <version>2.7.6</version>
</dependency>
```

**提交智能标签作业**

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class SubmitSmarttagJob {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        SubmitSmarttagJobRequest request = new SubmitSmarttagJobRequest();
        request.setInput("oss://xxx.mp4"); // 需要分析的文件OSS地址
        request.setPipelineId(""); // 管道ID，若填空，表示默认管道
        request.setTitle("<yourTitle>"); // 文件标题
        request.setContent("<yourContent>"); // 文件内容描述，选填
        request.setAcceptFormat(FormatType.JSON);
        try {
            // 接收请求结果response。
            SubmitSmarttagJobResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 查询智能标签作业

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class QuerySmarttagJob {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        QuerySmarttagJobRequest request = new QuerySmarttagJobRequest();
        request.setJobId("<JobId>");
        try {
            // 接收请求结果response。
            QuerySmarttagJobResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 添加模板

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class AddSmarttagTemplate {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        AddSmarttagTemplateRequest request = new AddSmarttagTemplateRequest();
        request.setIndustry("media");
        request.setScene("search");
        request.setAnalyseTypes("ocr,asr,classification,shows,face,role,object,tvstation,action,emotion,landmark,scene");
        request.setTemplateName("test");
        request.setFaceCategoryIds("politician,sensitive,celebrity");
        request.setIsDefault(false);
        try {
            // 接收请求结果response。
            AddSmarttagTemplateResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 查询模板

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class QuerySmarttagTemplateList {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        QuerySmarttagTemplateListRequest request = new QuerySmarttagTemplateListRequest();
        request.setTemplateId("6407");
        try {
            // 接收请求结果response。
            QuerySmarttagTemplateListResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 更新模板



```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class UpdateSmarttagTemplate {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        UpdateSmarttagTemplateRequest request = new UpdateSmarttagTemplateRequest();
        request.setTemplateName("update测试");
        request.setTemplateId("6407");
        try {
            // 接收请求结果response。
            UpdateSmarttagTemplateResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 删除模板

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class DeleteSmarttagTemplate {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        DeleteSmarttagTemplateRequest request = new DeleteSmarttagTemplateRequest();
        request.setTemplateId("6407");
        try {
            // 接收请求结果response。
            DeleteSmarttagTemplateResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 注册自定义人脸

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class RegisterCustomFace {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        RegisterCustomFaceRequest request = new RegisterCustomFaceRequest();
        request.setPersonId("<yourPersonId>");
        request.setImageUrl("http://xxxxx.jpeg");
        request.setCategoryId("<yourCategoryId>");
        try {
            // 接收请求结果response。
            RegisterCustomFaceResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 注销自定义人脸

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class UnregisterCustomFace {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        UnregisterCustomFaceRequest request = new UnregisterCustomFaceRequest();
        request.setPersonId("<yourPersonId>");
        request.setFaceId("<FaceId>");
        request.setCategoryId("<yourCategoryId>");
        try {
            // 接收请求结果response。
            UnregisterCustomFaceResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 添加自定义人物库或人物标签

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class TagCustomPerson {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        TagCustomPersonRequest request = new TagCustomPersonRequest();
        request.setCategoryId("<yourCategoryId>");
        request.setCategoryName("<yourCategoryName>");
        request.setCategoryDescription("<yourCategoryDescription>");
        try {
            // 接收请求结果response。
            TagCustomPersonResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 列出人物库所有人物和人脸信息

```
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.profile.DefaultProfile;
import com.google.gson.Gson;
import java.util.*;
import com.aliyuncs.mts.model.v20140618.*;
public class ListCustomPersons {
    public static void main(String[] args) {
        // 创建IAcsClient实例，需要补充AK信息和需要使用的服务区域，区域以beijing为例。
        DefaultProfile profile = DefaultProfile.getProfile("cn-beijing", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建request，并设置参数。
        ListCustomPersonsRequest request = new ListCustomPersonsRequest();
        request.setCategoryId("<yourCategoryId>");
        request.setPersonId("<yourPersonId>");
        try {
            // 接收请求结果response。
            ListCustomPersonsResponse response = client.getAcsResponse(request);
            System.out.println(new Gson().toJson(response));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            System.out.println("ErrCode:" + e.getErrCode());
            System.out.println("ErrMsg:" + e.getErrMsg());
            System.out.println("RequestId:" + e.getRequestId());
        }
    }
}
```

## 4.2. Python SDK

mts的Python SDK版本需要在2.7.6版本及以上。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  
...
```

## 提交智能标签作业

```
#!/usr/bin/env python  
#coding=utf-8  
from aliyunsdkcore.client import AcsClient  
from aliyunsdkcore.acs_exception.exceptions import ClientException  
from aliyunsdkcore.acs_exception.exceptions import ServerException  
from aliyunsdkmts.request.v20140618.SubmitSmarttagJobRequest import SubmitSmarttagJobRequest  
# 创建AcsClient实例。  
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')  
# 创建request，并设置参数。  
request = SubmitSmarttagJobRequest()  
request.set_accept_format('json')  
request.set_Title("<yourTitle>")  
request.set_Content("<yourContent>")  
request.set_PipelineId("")  
request.set_Input("oss://xxx.mp4")  
# 接收请求结果response。  
response = client.do_action_with_exception(request)  
# python2: print(response)  
print(str(response, encoding='utf-8'))
```

## 查询智能标签作业

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmmts.request.v20140618.QuerySmarttagJobRequest import QuerySmarttagJobRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = QuerySmarttagJobRequest()
request.set_accept_format('json')
request.set_JobId("<JobId>")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 添加模板



```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618.AddSmarttagTemplateRequest import AddSmarttagTemplateRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = AddSmarttagTemplateRequest()
request.set_accept_format('json')
request.set_Industry("media")
request.set_Scene("search")
request.set_AnalyseTypes("ocr,asr,classification,shows,face,role,object,tvstation,action,emotion,landmark,scene")
request.set_TemplateName("test")
request.set_FaceCategoryIds("politician,sensitive,celebrity")
request.set_IsDefault(False)
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 查询模板

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmmts.request.v20140618.QuerySmarttagTemplateListRequest import QuerySmarttagTemplateListRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = QuerySmarttagTemplateListRequest()
request.set_accept_format('json')
request.set_TemplateId("6407")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 更新模板

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmmts.request.v20140618.UpdateSmarttagTemplateRequest import UpdateSmarttagTemplateRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = UpdateSmarttagTemplateRequest()
request.set_accept_format('json')
request.set_TemplateName("update测试")
request.set_TemplateId("6407")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 删除模板

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618.DeleteSmarttagTemplateRequest import DeleteSmarttagTemplateRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = DeleteSmarttagTemplateRequest()
request.set_accept_format('json')
request.set_TemplateId("6407")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 注册自定义人脸

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618.RegisterCustomFaceRequest import RegisterCustomFaceRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = RegisterCustomFaceRequest()
request.set_accept_format('json')
request.set_PersonId("<yourPersonId>")
request.set_ImageUrl("http://xxxx.jpeg")
request.set_CategoryId("<yourCategoryId>")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 注销自定义人脸

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618.UnregisterCustomFaceRequest import UnregisterCustomFaceRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = UnregisterCustomFaceRequest()
request.set_accept_format('json')
request.set_PersonId("<yourPersonId>")
request.set_FacelId("<FacelId>")
request.set_CategoryId("<yourCategoryId>")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 添加自定义人物库或人物标签

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618.TagCustomPersonRequest import TagCustomPersonRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = TagCustomPersonRequest()
request.set_accept_format('json')
request.set_CategoryId("<yourCategoryId>")
request.set_CategoryName("<yourCategoryName>")
request.set_CategoryDescription("<yourCategoryDescription>")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 列出人物库所有人物和人脸信息

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkmts.request.v20140618.ListCustomPersonsRequest import ListCustomPersonsRequest
# 创建AcsClient实例。
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-beijing')
# 创建request，并设置参数。
request = ListCustomPersonsRequest()
request.set_accept_format('json')
request.set_CategoryId("<yourCategoryId>")
request.set_PersonId("<yourPersonId>")
# 接收请求结果response。
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding='utf-8'))
```

## 4.3. Php SDK

### 提交智能标签作业

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('SubmitSmarttagJob')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'Title' => "<yourTitle>",
                'Content' => "<yourContent>",
                'PipelineId' => "",
                'Input' => "oss://xxx.mp4",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 查询智能标签作业

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('QuerySmarttagJob')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'JobId' => "<JobId>",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 添加模板

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('AddSmarttagTemplate')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'Industry' => "media",
                'Scene' => "search",
                'AnalyseTypes' => "ocr,asr,classification,shows,face,role,object,tvstation,action,emotion
,landmark,scene",
                'TemplateName' => "test",
                'FaceCategoryIds' => "politician,sensitive,celebrity",
                'IsDefault' => "false",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 查询模板



```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('QuerySmarttagTemplatelist')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'TemplatelId' => "6407",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 更新模板

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('UpdateSmarttagTemplate')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'TemplateName' => "update测试",
                'TemplateId' => "6407",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 删除模板

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('DeleteSmarttagTemplate')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'TemplateId' => "6407",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 注册自定义人脸

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('RegisterCustomFace')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'PersonId' => "<yourPersonId>",
                'ImageUrl' => "http://xxxxx.jpeg",
                'CategoryId' => "<yourCategoryId>",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 注销自定义人脸

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('UnregisterCustomFace')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'PersonId' => "<yourPersonId>",
                'FaceId' => "<FaceId>",
                'CategoryId' => "<yourCategoryId>",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 添加自定义人物库或人物标签

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('TagCustomPerson')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'CategoryId' => "<yourCategoryId>",
                'CategoryName' => "<yourCategoryName>",
                'CategoryDescription' => "<yourCategoryDescription>",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```


## 列出人物库所有人物和人脸信息

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
// Download: https://github.com/aliyun/openapi-sdk-php
// Usage: https://github.com/aliyun/openapi-sdk-php/blob/master/README.md
AlibabaCloud::accessKeyClient('<accessKeyId>', '<accessSecret>')
    ->regionId('cn-beijing')
    ->asDefaultClient();
try {
    $result = AlibabaCloud::rpc()
        ->product('Mts')
        // ->scheme('https') // https | http
        ->version('2014-06-18')
        ->action('ListCustomPersons')
        ->method('POST')
        ->host('mts.cn-beijing.aliyuncs.com')
        ->options([
            'query' => [
                'RegionId' => "cn-beijing",
                'CategoryId' => "<yourCategoryId>",
                'PersonId' => "<yourPersonId>",
            ],
        ])
        ->request();
    print_r($result->toArray());
} catch (ClientException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
} catch (ServerException $e) {
    echo $e->getErrorMessage() . PHP_EOL;
}
```

## 5.播放器SDK

MPS播放器已和点播播放器合并，您可以直接参考点播提供的播放器。

参见 [播放器SDK产品介绍及使用说明](#)。

 说明 MPS用户可以使用新版提供的高级播放器进行升级。



## 6.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。

- [多地域支持](#)
  - [Java](#)
  - [PHP](#)
  - [Python](#)
- [Demo 工程](#)