

Alibaba Cloud ApsaraVideo for Media Processing SDK Reference

Issue: 20190417

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.








1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	It is used for commands.	Run the <code>cd / d C :/ windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid <i>Instance_ID</i></code>
[] or [a b]	It indicates that it is an optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
<code>{}</code> or <code>{a b}</code>	It indicates that it is a required value, and only one item can be selected.	<code>swich {stand slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 Create AccessKey.....	1
2 Upload SDK.....	2
2.1 Instructions for use.....	2
2.2 JavaScript version.....	5
2.3 Android version.....	7
2.4 iOS version.....	11
2.5 Upload SDK.....	15
3 Transcoding SDKs.....	18
3.1 SDK introduction.....	18
3.2 Multi-Region support.....	18
3.3 Java SDK.....	18
3.3.1 Overview.....	18
3.3.2 Installation.....	19
3.3.3 Quick start.....	21
3.3.4 Transcoding.....	22
3.3.5 Watermarks.....	25
3.3.6 Screenshot.....	29
3.3.7 HLS data encryption.....	33
3.3.8 Create HLS standard encryption workflow.....	35
3.3.9 MPS queues management.....	38
3.3.10 Query media - Use OSS file address.....	39
3.3.11 Add media.....	41
3.3.12 Splicing and simple cutting.....	42
3.3.13 Add media workflow.....	46
3.3.14 Splicing-opening and ending scenes.....	49
3.4 Python SDK.....	53
3.4.1 Overview.....	53
3.4.2 Installation.....	54
3.4.3 Quick start.....	54
3.4.4 Transcoding.....	55
3.4.5 Watermarks.....	57
3.4.6 Screenshot.....	61
3.4.7 Query media - Use OSS file address.....	63
3.4.8 Create HLS standard encryption workflow.....	63
3.4.9 Splicing and simple cutting.....	65
3.4.10 Splicing-opening and ending scenes.....	67
3.4.11 Add media.....	70
3.4.12 Add media workflow.....	70

3.5 PHP SDK.....	72
3.5.1 Overview.....	72
3.5.2 Installation.....	73
3.5.3 Quick start.....	73
3.5.4 Transcoding.....	74
3.5.5 Watermarks.....	77
3.5.6 Screenshot.....	81
3.5.7 Splicing and simple cutting.....	83
3.5.8 Splicing-opening and ending scenes.....	85
3.5.9 Create HLS standard encryption workflow.....	88
3.5.10 MPS queue management.....	91
3.5.11 Query media - Use OSS file address.....	92
3.5.12 Add media.....	93
3.5.13 Add media workflow.....	94
4 Demo project.....	96

1 Create AccessKey

Perform the following steps to create AccessKey.

1. Log on to the Alibaba Cloud website.
2. Visit the [AccessKey console](#).
3. Create and manage your AccessKey.

2 Upload SDK

2.1 Instructions for use

The upload SDK provides the file list management and upload control functions. File list management allows you to add, delete, cancel, resume, retrieve, and clear files. Upload control allows you to start, stop, pause, and resume file upload. The SDK provides callback events to monitor the status and progress changes during upload.

Upload process

Initialize the upload SDK > Select a file > Add the file to the list > Start upload > Upload completion events.

- Initialize the upload SDK

The AccessKey and token can be used to grant permissions during initialization. Considering the security of the AccessKey stored on the client, it is recommended that the AccessKey be used only for testing and the token be used in the production environment. For more information, see [Developer guide > Upload a video file > Overview](#).

- Select a file

Select a local file to be uploaded.

- Add the file to the list

Add all files to be uploaded to the list using `addFile`.

- Start upload.

Call `start` to start the upload process.

- Upload completion events

The events include `OnUploadSucceed` and `OnUploadFailed`.

Concepts and descriptions

- Multipart upload and status

The SDK uses the multipart upload mode, in which the status is valid only for one execution. If the app exits due to a specific reason (for example, shutdown, closing the browser page, closing the app, or abnormal app exit), the file must be uploaded again.

- **Authorization expiration**

The token is only valid for a certain period. After the token expires, the upload process is interrupted and cannot be automatically resumed. To resume upload, obtain a new token from the backend and call the `resumeUploadWithAuth` function

- **Switching between a 3G/4G network and a Wi-Fi network on the mobile end**

To avoid traffic waste on 3G/4G networks, when the app switches to a 3G/4G network, it must automatically detect the network and call `pause` to suspend upload. After the app detects that a Wi-Fi network is used, it calls `resume` to resume upload.

- The SDKs of the following three types of terminals are provided:

- **HTML5**: It can be integrated to PC browsers. The development language is JavaScript.
- **iOS**: It can be integrated to iOS apps in the language of Object-C.
- **Android**: It can be integrated to Android apps in the language of Java.

Features

File list management

API	Description
<code>addFile</code>	Adds a file to the list. Files are uploaded in order of addition.
<code>deleteFile</code>	Deletes a file from the list.
<code>cancelFile</code>	Cancels a file in the list but does not delete the file from the list. After this API is used, upload of this file is skipped (not supported by the JavaScript version).
<code>resumeFile</code>	Resumes the status of the file that is canceled in the list (not supported by the JavaScript version). This API is not used to resume upload.
<code>listFiles</code>	Obtains the list.
<code>clearFiles</code>	Clears the list. After this API is used, upload stops and files, even the files being uploaded, are cleared (not supported by the JavaScript version).

Upload control

API	Description
start	Starts upload.
stop	Stop uploading
pause	Stops upload (JavaScript version is not supported).
resume	Pauses upload (not supported by the JavaScript version).
resumeUploadWithToken	Uses a new token to resume upload after the existing token expires.

Callback events

Event	Description
OnUploadStarted	This event is triggered when upload of each file starts.
OnUploadSucceed	Upload succeeds.
OnUploadFailed	Upload fails. If resumable errors occur , for example the network has an exception or times out, the file upload can be resumed. If non-resumable errors occur, for example the upload credential is incorrect or the file does not exist, the file upload fails.
OnUploadProgress	The upload progress report. This event is triggered when multipart upload succeeds.
OnUploadTokenExpired	The token times out. To resume upload , obtain a new token from the server and call the resumeUploadWithToken function.
OnUploadRetry	This event is triggered when the status switches from normal to abnormal during upload For example, the network has an exception or times out (not supported by the JavaScript version).

Event	Description
OnUploadRetryResume	This event is triggered when the status is resumed from abnormal to normal during upload (not supported by the JavaScript version).

2.2 JavaScript version

Installation

[Download the upload SDK](#)

Introduce the following two JavaScript scripts on the page:

```
< script src = " aliyun - sdk . min . js "></ script >
< script src = " vod - sdk - upload - 1 . 0 . 6 . min . js "></
script >
```

Initialization: create a VODUpload instance.

Set the callback function.

```
var uploader = new VODUpload ({
  // Starts upload
  'onUploadStarted': function (uploadInfo) {
    log (" onUploadSt arted :" + uploadInfo . file . name + ",
endpoint :" + uploadInfo . endpoint + ", bucket :" + uploadInfo .
bucket + ", object :" + uploadInfo . object );
  }
  // File upload succeeds
  'onUploadSucceed': function (uploadInfo) {
    log (" onUploadSu cceed :" + uploadInfo . file . name + ",
endpoint :" + uploadInfo . endpoint + ", bucket :" + uploadInfo .
bucket + ", object :" + uploadInfo . object );
  },
  // File upload fails
  'onUploadFailed': function (uploadInfo , code , message )
{
  log (" onUploadFa iled : file :" + uploadInfo . file . name
+ ", code :" + code + ", message :" + message );
},
  // The file upload progress , in bytes
  'onUploadProgress': function (uploadInfo , totalSize ,
uploadedSize ) {
  log (" onUploadPr ogress : file :" + uploadInfo . file .
name + ", fileSize :" + totalSize + ", percent :" + Math . ceil
( uploadedSi ze * 100 / totalSize ) + "%");
},
  // Token expires
  'onUploadTokenExpired': function () {
    console . log (" onUploadTo kenExpired ");
    // uploader . resumeUplo adWithToke n ( accessKeyI d ,
accessKeyS ecret , secretToke n , expireTime );
  }
}
```

```
});
```

List management

- Add a file to be uploaded



Note:

The file size cannot exceed 10 GB.

Use the standard input mode for selecting a file.

```
< form    action ="">
< input  type =" file "  name =" file "  id =" files "  multiple />
</ form >
userData = '';
document . getElement ById (" files ")
    . addEventLi stener (' change ', function ( event ) {
    for ( var    i = 0 ; i < event . target . files . length ;
i ++ ) {
        // The    logic    code
    }
    });
```

Obtain the selected file and add it to the upload list.

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



Note:

- `event.target.files[i]`: file list that user select,
- `endpoint`: endpoint of OSS,
- `bucket`: bucket of OSS,
- `object`: object of OSS,
- `userData`: the last parameter of `addFile`, a json object.

During uploading, obtain the attributes (the title, tag, description, category, and custom data) of a media set in the following way: The last parameter `userData` of the `addFile` function is a JSON object. The first-level VOD is required, and VOD contains the five attributes. Example:

```
var    userData = '{" Vod ":{" Title ":" I am the title ",
" Descriptio n ":" I am the descriptio n
",
" Cateoid ": " 1 ",
" Tags ":" tag1 , tag2 , tag3 ",
```



```
" UserData ":" user  data "}}';
```

- **Delete the uploaded file.** `index` corresponds to the index of the elements in the list returned by `listFiles`.

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

- **Cancel upload of a single file.**

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

- **Resume upload of a single file.**

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

- **Obtain the upload file list.**

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

- **Clear the upload file list.**

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

Upload control

- **Start upload.**

```
uploader . startUploa d ();
```

- **Stop upload**

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

- **Resume upload after the upload credential becomes invalid.**

```
uploader . resumeUploa dWithToke n ( accessKeyI d , accessKeyS  
ecret , secretToke n , expireTime );
```

2.3 Android version

Environment requirements

Android 2.3 or a later version

Installation

[OSS Android SDK](#)

[Upload SDK](#)

- Directly introduce the JAR package

After you download the ZIP package of the VODUpload Android SDK, perform the following steps (applicable to Android Studio or Eclipse):

- Decompress the SDK to obtain the following JAR packages in the libs directory: aliyun-oss-sdk-android-xxx.jar, okhttp-2.7.0.jar, okio-2.6.0.jar, and aliyun-vod-upload-android-sdk-xxx.jar.
- Import the four JAR packages to the libs directory of the project.
- Set permissions

The following are the Android permissions required by the VODUpload Android SDK. Make sure that these permissions are already set in your AndroidManifest.xml file. Otherwise, the SDK cannot work normally.

```
< uses - permission    android : name =" android . permission .
INTERNET "></ uses - permission >
< uses - permission    android : name =" android . permission .
ACCESS_NET_WORK_STATE "></ uses - permission >
< uses - permission    android : name =" android . permission .
ACCESS_WIFI_STATE "></ uses - permission >
< uses - permission    android : name =" android . permission .
WRITE_EXTERNAL_STORAGE "></ uses - permission >
```

Create a VODUpload instance

Set the callback function.

```
VODUploadC allback callback = new VODUploadC allback () {
    /**
     * Triggered when the file upload starts
     */
    void onUploadStarted () {}
    /**
     * Callback after successful upload
     */
    void onUploadSucceed ( UploadFile Info info ) {}
    /**
     * Upload failed
     */
    void onUploadFailed ( UploadFile Info info , String
code , String message ) {}
    /**
     * Callback upload progress
     * @param uploadedSize The number of uploaded bytes
     * @param totalSize The total number of required
bytes
```

```

    */
    void onUploadProgress ( UploadFile Info info , long
    uploadedSize , long totalSize ) {;}
    /**
    * This API is called back after the upload
    credential expires
    * Obtain a new upload credential in the callback
    and call resumeUploadWithAuth to continue upload
    */
    void onUploadTokenExpired () {;}
    /**
    * Triggered when the status switches from normal to
    abnormal during upload
    */
    void onUploadRetry ( String code , String message ) {;}
    /**
    * Triggered when the status is resumed from
    abnormal during upload
    */
    void onUploadRetryResume () {;}
};
VODUploadClient uploader = new VODUploadClientImpl (
getContext ());

```

Initialization

Enter the authorization information in either of the following ways:

- AccessKey

It is simple but not safe. It is recommended that this way be used in the test environment.

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

- Token

It is safe but complex. It is recommended that this way be used in the production environment. A token is temporary and valid for a period. Therefore, sending a token is safe.

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

List management

- Add a file to be uploaded.



Note:

The file size cannot exceed 4 GB.

```
uploader . addFile ("< uploadFile Path >",
"< endpoint >", // For example , the Hangzhou
region " http :// oss - cn - hangzhou . aliyuncs . com "
```

```
name "< bucketName >", // Enter the actual bucket
      "< objectKey >");
```

During uploading, obtain the attributes (the title, tag, description, category, cover URL, and custom data) of a media set in the following way: `addFile` contains a reload function, in which the last parameter is a `VodInfo` object. The definitions are as follows:

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

```
private String coverUrl ;
```

- **Delete the uploaded file.** `index` corresponds to the index of the elements in the list returned by `listFiles`.

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

- **Cancel upload of a single file in the list.**

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

- **Resume upload of a single file in the list.**

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

- **Obtain the upload file list.**

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

- **Clear the upload file list.**

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

Upload control

- **Start upload.**

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

- **Stop upload.**

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

- **Pause upload.**

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

- **Resume upload.**

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

- **Resume upload after the token is invalid.**

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

2.4 iOS version

Environment requirements

iOS 7.0 or a later version

Installation

[OSS iOS SDK](#)

[Download the upload SDK](#)

- Directly introduce the frameworks

Introduce the frameworks of the OSS iOS SDK and VODUpload iOS SDK.

In Xcode, drag the frameworks and drop them to your target, and select “Copy items if needed” in the displayed dialog box.

- Introduce the header file to your project

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



Note:

After you introduce the frameworks, add `-ObjC` to Other Linker Flags of Build Settings in your project. If the `-force_load` option has been configured for your project, add `-force_load <framework path >/AliyunOSSiOS`.

- Compatible with IPv6-Only networks.

The OSS mobile SDK has introduced the HTTPDNS for domain name resolution to avoid domain name resolution hijacking in a wireless network and directly uses IP addresses for requests to the server. In an IPv6-Only network, compatibility issues may occur. The app officially issued the review requirements for apps, requiring apps to be IPv6-only network compatible. To this end, the SDK starts to be compatible from `V2.5.0`. In the new version, apart from `-ObjC` settings, two system libraries must be introduced:

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

Create a VODUpload instance

Set the callback function.

```
OnUploadStartedListener testUploadStartedCallbackFunc = ^(
UploadFileInfo * fileInfo) {
    NSLog(@"upload started.");
};
OnUploadSuccessListener testSuccessfulCallbackFunc = ^(
NSString * filePath){
    NSLog(@"file :%@ upload success!", filePath);
};
```

```

    OnUploadFailedListener testFailed CallbackFunc = ^(
    NSString * filePath , NSString * code , NSString * message ){
        NSLog(@" failed code = %@, error message = %@", code ,
        message );
    };
    // Unit : byte
    OnUploadProgressListener testProgressCallbackFunc = ^(
    NSString * filePath , long uploadedSize , long totalSize ) {
        NSLog(@" progress uploadedSize : %li , totalSize : %li
        ", uploadedSize , totalSize );
    };
    OnUploadTokenExpiredListener testTokenExpiredCallbackFunc
    = ^{
        NSLog(@"* token expired.");
        // get token and call resumeUploadWithAuth .
    };
    OnUploadRetryListener testUploadRetryListener = ^{
        NSLog(@" retry begin .") ;
    };
    OnUploadRetryResumeListener testUploadRetryResumeListener
    = ^{
        NSLog(@" retry resume .") ;
    };
    VODUploadListener * listener ;
    listener = [[ VODUploadListener alloc ] init ];
    listener . started = testUploadStartedCallbackFunc ;
    listener . success = testSuccessfulCallbackFunc ;
    listener . failure = testFailed CallbackFunc ;
    listener . progress = testProgressCallbackFunc ;
    listener . expire = testTokenExpiredCallbackFunc ;
    listener . retry = testUploadRetryListener ;
    listener . retryResume = testUploadRetryResumeListener ;

```

Initialize the upload SDK

Enter the authorization information in either of the following ways:

- **AccessKey**

It is simple but not safe. It is recommended that this way be used in the test environment.

```

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

```

- **Token**

It is safe but complex. It is recommended that this way be used in the production environment. A token is temporary and valid for a period. Therefore, sending a token is safe.

```

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

```

```
listener : listener ];
```

- **List Management**

- Add a file to be uploaded.



Note:

The file size cannot exceed 4 GB.

```
[ uploader  addFile :< uploadFile  Path >
             endpoint :< endpoint > // Example : ' http :// oss - cn -
             hangzhou . aliyuncs . com '
             bucket :< bucketName > // Enter  the  actual  bucket
             name
             object :< objectKey >];
```

During uploading, obtain the attributes (the title, tag, description, category, cover URL, and custom data) of a media set in the following way: `addFile` contains a reload function, in which the last parameter is a `VodInfo` object. The definitions are as follows:

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



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

- **Delete the uploaded file.**

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

- **Cancel upload of a single file in the list.**

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

- **Resume upload of a single file in the list.**

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

- **Obtain the upload file list.**

```
[ uploader listFiles ];
```

- **Clear the upload file list.**

```
[ uploader clearFiles ];
```

Upload control

- **Start upload.**

```
[ uploader start ];
```

- **Stop upload.**

```
[ uploader stop ];
```

- **Pause upload.**

```
[ uploader pause ];
```

- **Resume upload.**

```
[ uploader resume ];
```

- **Resume upload after the token is invalid.**

```
[ uploader resumeWith Token :< accessKeyId >
    accessKeySecret :< accessKeySecret >
    secretToken :< secretToken >
    expireTime :< expireTime >]
```

2.5 Upload SDK

Latest versions

- [SDK for JavaScript 1.0.6, sample code](#)
- [SDK for iOS 1.0.7, sample code](#)

- [SDK for Android 1.0.6, sample code](#)

1.0.7

- [SDK for iOS 1.0.7](#)

Updates

- **Added support for bitcode**

1.0.6

- [SDK for JavaScript 1.0., sample code](#)
- [SDK for iOS 1.0.6, sample code](#)
- [SDK for Android 1.0.6, sample code](#)

Updates:

- For Android and iOS, index serial numbers, instead of file names, are used in the parameters of list management APIs (cancelFile, deleteFile, and resumeFile) to correspond to listFiles.
- Both the on-demand and traditional OSS upload modes are supported.
- The on-demand upload mode is added in the sample code.

1.0.5

- [SDK for JavaScript 1.0.5, sample code](#)
- [SDK for iOS 1.0.5, sample code](#)
- [SDK for Android 1.0.5, sample code](#)

Updates

- The pause, resuming, and network exceptions are added.
- The title, tag, cover URL, category ID, description, and custom data of the media set can be specified for a video to be uploaded.
- The SDK for iOS integrates OSS 2.6.0 and supports the App Transport Security (ATS) standard of Apple Inc.

0.0.4

- [SDK for JavaScript 0.0., sample code](#)
- [SDK for iOS 0.0.4, sample code](#)
- [SDK for Android 0.0.4, sample code](#)

Updates

- First release

3 Transcoding SDKs

3.1 SDK introduction

SDK introduction details the installation, application, and example codes for MPS SDK.

MPS SDK is based on Alibaba Cloud SDK. For more information about Alibaba Cloud SDK, see:

- [Alibaba Cloud SDK](#)
- [Alibaba Cloud API](#)

3.2 Multi-Region support

MPS is available in many regions. For more information about regions list and RegionId corresponding to SDK, see:

- [MPS service regions](#)
- [RegionId corresponding to regions](#)

3.3 Java SDK

3.3.1 Overview

MPS Java SDK is based on Alibaba Cloud Java SDK. This article introduces the basic knowledge about Alibaba Cloud Java SDK.

- [Alibaba Cloud SDK Developer Guide > Java SDK > Getting started](#)
- [Alibaba Cloud SDK Developer Guide > Java SDK > Use Java SDK](#)
- [Alibaba Cloud Java SDK GitHub](#)

After learning about the basic knowledge, you can perform MPS Java SDK installation. For more information, see [MPS > SDK Reference > Java SDK > Installation](#).

3.3.2 Installation

This article introduces Maven installation method recommended by Alibaba Cloud Java SDK. Specifically, it includes two steps. First, add the Maven repository of Alibaba Cloud Java SDK to the pom.xml configuration file. Then, add MPS dependency.

1. Add a Maven repository.

```
< 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. Add dependency.

Alibaba Java SDK Core and MPS Java SDK detailed version:

- [Alibaba Cloud Java SDK Core](#)
- [MPS Java SDK](#)

Take the core of V3.5.0 SDK Core and V2.5.2 MPS SDK as an example:

```
< dependency >
  < groupId > com . aliyun </ groupId >
  < artifactId > aliyun - java - sdk - core </ artifactId >
  < version > 3 . 5 . 0 </ version >
</ dependency >
< dependency >
  < groupId > com . aliyun </ groupId >
  < artifactId > aliyun - java - sdk - mts </ artifactId >
  < version > 2 . 5 . 2 </ version >
</ dependency >
```

In addition, a json repository dependency optional. In MPS API, many parameters are defined through json. And many json repositories for java are available, you can select the repository that you are familiar with. Take V1.2.46 [fastjson](#) as an example:

```
< dependency >
  < groupId > com . alibaba </ groupId >
  < artifactId > fastjson </ artifactId >
  < version > 1 . 2 . 46 </ version >
```

```
</ dependency >
```

pom.xml example:

```
<? xml version="1.0" encoding="UTF-8" ? >
  < project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd" >
    < modelVersion > 4.0.0 </ modelVersion >
    < groupId > mps - demo - project </ groupId >
    < artifactId > mps - demo - project </ artifactId >
    < version > 0.0.1 - SNAPSHOT </ version >
    < repositories >
      < repository >
        < id > sonatype - nexus - staging </ id >
        < name > Sonatype Nexus Staging </ name >
        < url > https://oss.sonatype.org/service/local/staging/
          deploy/maven2 /</ url >
        < releases >
          < enabled > true </ enabled >
        </ releases >
        < snapshots >
          < enabled > true </ enabled >
        </ snapshots >
      </ repository >
    </ repositories >
    < dependencies >
      < dependency >
        < groupId > com . aliyun </ groupId >
        < artifactId > aliyun - java - sdk - core </ artifactId >
        < version > 3 . 5 . 0 </ version >
      </ dependency >
      < dependency >
        < groupId > com . aliyun </ groupId >
        < artifactId > aliyun - java - sdk - mts </ artifactId >
        < version > 2 . 5 . 2 </ version >
      </ dependency >
      < dependency >
        < groupId > com . alibaba </ groupId >
        < artifactId > fastjson </ artifactId >
        < version > 1 . 2 . 46 </ version >
      </ dependency >
    </ dependencies >
    < build >
      < finalName > ${ artifactId } - ${ version } </ finalName >
      < plugins >
        < plugin >
          < groupId > org . apache . maven . plugins </ groupId >
          < artifactId > maven - compiler - plugin </ artifactId >
          < version > 2 . 3 . 2 </ version >
          < configuration >
            < source > 1 . 6 </ source >
            < target > 1 . 6 </ target >
          </ configuration >
        </ plugin >
      </ plugins >
    </ build >
```

```
</ project >
```

3.3.3 Quick start

This article describes the Java SDK quick start process.

1. Create AcsClient instance.

```
DefaultProfile profile = DefaultProfile.getProfile (
    mpsRegionId, // region ID
    accountId, // AccessKey ID of RAM
    accessKeySecret); // Access Key Secret of
IAcsClient client = new DefaultAcsClient ( profile );
```

2. Create request, and set parameters.

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

3. Initiate API request, and display the return value.

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

Full code

```
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);
            // Create DefaultAcsClient instance and finish
            initialization
            DefaultProfile profile = DefaultProfile .
getProfile (
                mpsRegionId, // region ID
                accountId, // AccessKey ID of
                accessKeySecret); // Access Key Secret
            of RAM account
```

```

IAcsClient client = new DefaultAcsClient (
profile );
// Create API request and set parameters
SearchPipe lineRequest request = new SearchPipe
lineRequest ();
// initiate request and handle response or
exceptions
SearchPipe lineResponse 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 Transcoding

1. Create AcsClient instance.

```

DefaultProfile profile = DefaultProfile . getProfile (
    mpsRegionId , // region ID
    accountId , // AccessKey ID of RAM
    accessKeySecret ); // Access Key Secret of
RAM account
IAcsClient client = new DefaultAcsClient ( profile );

```

2. Create request, and set parameters.

```

SubmitJobsRequest request = new SubmitJobsRequest ();

```

3. Transcoding parameters.

• Input

```

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

```

• 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. Initiate API request and display returned value.

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

Full codes

```
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 . ClientExce ption ;
import com . aliyuncs . exceptions . ServerExce ption ;
import com . aliyuncs . mts . model . v20140618 . * ;
public class SimpleTran scode {
    private static String accessKeyI d = " xxx " ;
    private static String accessKeyS ecret = " xxx " ;
    private static String mpsRegionI d = " cn - hangzhou " ;
    private static String pipelineI d = " xxx " ;
    private static String templateI d = " S000000001 - 200010 " ;
    private static String ossLocatio n = " oss - cn -
hangzhou " ;
    private static String ossBucket = " xxx " ;
    private static String ossInputOb ject = " input . mp4 " ;
    private static String ossOutputO bject = " output . mp4
";
    public static void main ( String [] args ) {
        // Create DefaultAcs Client instance and complete
        // initialization
        DefaultPro file profile = DefaultPro file . getProfile
(
            mpsRegionI d , // region D
            accessKeyI d , // AccessKey ID of RAM
            accessKeyS ecret ); // Access Key Secret of
RAM account
        IAcsClient client = new DefaultAcs Client ( profile );
        // Create API request and set parameters
        SubmitJobs Request request = new SubmitJobs Request
();
        // Input
        JSONObject input = new JSONObject ();
        input . put ( " Location " , ossLocatio n );
        input . put ( " Bucket " , ossBucket );
        try {
            input . put ( " Object " , URLEncoder . encode (
            ossInputOb ject , " utf - 8 " ));
        } catch ( Unsupporte dEncodingE xception e ) {
            throw new RuntimeExc eption ( " input URL encode
            failed " );
        }
        request . setInput ( input . toJSONStri ng ());
        // Output
        String outputOSSO bject ;
        try {
            outputOSSO bject = URLEncoder . encode ( ossOutputO
            bject , " utf - 8 " );
        } catch ( Unsupporte dEncodingE xception e ) {
            throw new RuntimeExc eption ( " output URL encode
            failed " );
        }
        JSONObject output = new JSONObject ();
        output . put ( " OutputObje ct " , outputOSSO bject );
        // Ouput -> Container
        JSONObject container = new JSONObject ();
        container . put ( " Format " , " mp4 " );
        output . put ( " Container " , container . toJSONStri ng ());
        // Ouput -> Video
        JSONObject video = new JSONObject ();
        video . put ( " Codec " , " H . 264 " );
        video . put ( " Bitrate " , " 1500 " );
        video . put ( " Width " , " 1280 " );
        video . put ( " Fps " , " 25 " );
        output . put ( " Video " , video . toJSONStri ng ());
        // 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 ());
// Output -> 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 );
// Initiate request and handle response or
exceptions
SubmitJobs Response 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 ();
}
}
}
}

```

3.3.5 Watermarks

1. Create AcClient instance.

```

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

```

2. Create request, and set parameters.

```

SubmitJobs Request request = new SubmitJobs Request ();

```

3. Set transcoding parameters.

- Image watermark

```

// Image Watermark
JSONObject imageWatermarkInput = new JSONObject ();

```

```

        imageWater markInput . put ( " Location ", ossLocatio n );
        imageWater markInput . put ( " Bucket ", ossBucket );
        try {
            imageWater markInput . put ( " Object ", URLEncoder .
encode ( imageWater markObject , " utf - 8 "));
        } catch ( Unsupporte dEncodingE xception e ) {
            throw new RuntimeExc eption ( " imageWatet mark
Input URL encode failed " );
        }
        JSONObject imageWater mark = new JSONObject ( );
        imageWater mark . put ( " WaterMarkT emplateId ",
watermarkT emplateId );
        imageWater mark . put ( " Type ", " Image " );
        imageWater mark . put ( " InputFile ", imageWater markInput
);
        imageWater mark . put ( " ReferPos ", " TopRight " );
        imageWater mark . put ( " Width ", " 0 . 05 " );
        imageWater mark . put ( " Dx ", " 0 " );
        imageWater mark . put ( " Dy ", " 0 " );

```

- **Text watermark**

```

// Text Watermark
JSONObject textConfig = new JSONObject ( );
textConfig . put ( " Content ", " 5rWL6K + V5paH5a2X5 rC05Y2w
");
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 textWaterm ark = new JSONObject ( );
textWaterm ark . put ( " WaterMarkT emplateId ", watermarkT
emplateId );
textWaterm ark . put ( " Type ", " Text " );
textWaterm ark . put ( " TextWaterM ark ", textConfig .
toJSONStri ng ( ));

```

- **Video watermark**

```

// Video Watermark
JSONObject videoWater markInput = new JSONObject ( );
videoWater markInput . put ( " Location ", ossLocatio n );
videoWater markInput . put ( " Bucket ", ossBucket );
try {
    videoWater markInput . put ( " Object ", URLEncoder .
encode ( videoWater markObject , " utf - 8 "));
} catch ( Unsupporte dEncodingE xception e ) {
    throw new RuntimeExc eption ( " videoWatet mark
Input URL encode failed " );
}
JSONObject videoWater mark = new JSONObject ( );
videoWater mark . put ( " WaterMarkT emplateId ",
watermarkT emplateId );
videoWater mark . put ( " Type ", " Image " );
videoWater mark . put ( " InputFile ", videoWater markInput
);
videoWater mark . put ( " ReferPos ", " BottomLeft " );
videoWater mark . put ( " Height ", " 240 " );
videoWater mark . put ( " Dx ", " 0 " );

```

```
videoWater mark . put ( " Dy ", " 0 " );
```

4. Initiate API request and display returned value.

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

Full codes

```
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 . DefaultAcs Client ;
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 ) {
        // DefaultAcs Client
        DefaultProfile profile = DefaultProfile . getProfile
(
            mpsRegionId , // Region ID
            accessKeyId , // AccessKey ID
            accessKeySecret ) ; // Access Key Secret
        IAcsClient client = new DefaultAcs Client ( profile ) ;
        // request
        SubmitJobs Request request = new SubmitJobs Request
() ;
        // Input
        JSONObject input = new JSONObject ( ) ;
        input . put ( " Location " , ossLocation ) ;
```

```

        input . put ( " Bucket ", ossBucket );
        try {
            ossInputObject input . put ( " Object ", URLEncoder . encode (
                " 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 + V5paH5a2X5 rC05Y2w
        " );
        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 {

```

```

        videoWater markInput . put ( " Object ", URLEncoder .
encode ( videoWater markObject , " utf - 8 "));
    } catch ( Unsupporte dEncodingE xception e ) {
        throw new RuntimeExc eption ( " videoWatet mark
Input URL encode failed " );
    }
    JSONObject videoWater mark = new JSONObject ( );
    videoWater mark . put ( " WaterMarkT emplateId ",
watermarkT emplateId );
    videoWater mark . put ( " Type ", " Image " );
    videoWater mark . put ( " InputFile ", videoWater markInput
);
    videoWater mark . put ( " ReferPos ", " BottomLeft " );
    videoWater mark . put ( " Height ", " 240 " );
    videoWater mark . put ( " Dx ", " 0 " );
    videoWater mark . put ( " Dy ", " 0 " );
    // Output -> Watermarks
    JSONArray watermarks = new JSONArray ( );
    watermarks . add ( imageWater mark );
    watermarks . add ( textWaterm ark );
    watermarks . add ( videoWater mark );
    output . put ( " WaterMarks ", watermarks . toJSONStri ng
());
    // Outputs
    JSONArray outputs = new JSONArray ( );
    outputs . add ( output );
    request . setOutputs ( outputs . toJSONStri ng ( ));
    request . setOutputB ucket ( ossBucket );
    request . setOutputL ocation ( ossLocatio n );
    // PipelineId
    request . setPipelin eId ( pipelineId );
    // call api
    SubmitJobs Response response ;
    try {
        response = client . getAc sResp onse ( request );
        System . out . println ( " RequestId is : "+ response .
getRequest Id ( ));
        if ( response . getJobResu ltList ( ). get ( 0 ) .
getSuccess ( )) {
            System . out . println ( " JobId is : "+ response .
getJobResu ltList ( ). get ( 0 ) . getJob ( ). getJobId ( ));
        } else {
            System . out . println ( " SubmitJobs Failed code
:" + response . getJobResu ltList ( ). get ( 0 ) . getCode ( ) +
" message : " + response .
getJobResu ltList ( ). get ( 0 ) . getMessage ( ));
        }
    } catch ( ServerExce ption e ) {
        e . printStack Trace ( );
    } catch ( ClientExce ption e ) {
        e . printStack Trace ( );
    }
}
}
}

```

3.3.6 Screenshot

1. Create an AcsClient instance.

```

DefaultPro file profile = DefaultPro file . getProfile (
    mpsRegionI d , // Region ID
    accessKeyI d , // AccessKey ID

```

```
        accessKeySecret ); // Access Key Secret
IAcsClient client = new DefaultAcsClient ( profile );
```

2. Create a request and set parameters.

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

3. Set the screenshot parameters.

• Input

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

• SnapshotConfig

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

- OutputFile

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

- Time

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

- Interval/Num

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



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

- **Width/Height**

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

4. Initiate an API request and display the response value.

```
SubmitSnapshotJobResponse response ;
response = client . getAcsResponse ( request );
System . out . println ( " RequestId is :"+ response .
getRequestId ());
System . out . println ( " JobId is :"+ response . getSnapshotJob ( ) .
getJobId ());
System . out . println ( String . format (
" http ://% s .% s .
aliyuncs . com / output_000_01 . jpg ",
ossBucket ,
ossLocation ));
System . out . println ( String . format (
" http ://% s .% s .
aliyuncs . com / output_000_02 . jpg ",
ossBucket ,
ossLocation ));
System . out . println ( String . format (
" http ://% s .% s .
aliyuncs . com / output_000_03 . jpg ",
ossBucket ,
ossLocation ));
```

Full code

```
package com . aliyun . mts ;
import java . io . UnsupportedEncodingException ;
import java . net . URLDecoder ;
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
SubmitSnap shotJobReq uest request = new SubmitSnap
shotJobReq uest ();
// Input
JSONObject input = new JSONObject ();
input . put ( " Location ", ossLocatio n );
input . put ( " Bucket ", ossBucket );
try {
input . put ( " Object ", URLEncoder . encode (
ossInputOb ject , " utf - 8 "));
} catch ( Unsupporte dEncodingE xception e ) {
throw new RuntimeExc eption ( " input URL encode
failed " );
}
request . setInput ( input . toJSONStri ng ());
// SnapshotCo nfig
JSONObject snapshotCo nfig = new JSONObject ();
// SnapshotCo nfig -> OutputFile
JSONObject output = new JSONObject ();
output . put ( " Location ", ossLocatio n );
output . put ( " Bucket ", ossBucket );
try {
output . put ( " Object ", URLEncoder . encode (
ossOutputO bject , " utf - 8 "));
} catch ( Unsupporte dEncodingE xception e ) {
throw new RuntimeExc eption ( " output URL encode
failed " );
}
snapshotCo nfig . put ( " OutputFile ", output . toJSONStri
ng ());
// SnapshotCo nfig -> Time
snapshotCo nfig . put ( " Time ", " 2 " );
// SnapshotCo nfig -> Interval / Num
snapshotCo nfig . put ( " Interval ", " 2 " );
snapshotCo nfig . put ( " Num ", " 3 " );
// SnapshotCo nfig -> Width / Height
snapshotCo nfig . put ( " Height ", " 360 " );
// SnapshotCo nfig
request . setSnapsho tConfig ( snapshotCo nfig .
toJSONStri ng ());
// PipelineId
request . setPipelid ( pipelineId );
// call api
SubmitSnap shotJobRes ponse response ;
try {
response = client . getAc sResp onse ( request );
System . out . println ( " RequestId is : "+ response .
getRequest
Id ());
System . out . println ( " JobId is : "+ response .
getSnapsho
tJob ( ) . getId ());
System . out . println ( String . format (
" http ://% s . % s . aliyuncs
. com / output_000 01 . jpg ",
ossBucket ,
ossLocatio n ));
System . out . println ( String . format (
" http ://% s . % s . aliyuncs
. com / output_000 02 . jpg ",
ossBucket ,
ossLocatio n ));
System . out . println ( String . format (
" http ://% s . % s . aliyuncs
. com / output_000 03 . jpg ",
ossBucket ,

```

```

        ossLocation));
    } catch ( ServerException e ) {
        e.printStackTrace();
    } catch ( ClientException e ) {
        e.printStackTrace();
    }
}
}
}

```

3.3.7 HLS data encryption

Scenario

- HLS standard data encryption applies to "protect video protection", it can prevent illegal downloads and illegal dissemination.

Limits

- To use HLS standard data encryption, you must use SubmitJobs API.
- HLS standard data encryption function cannot be used in workflow.

Code example

- For more information about MPS SDK, [Installation](#).
- Other code examples

```

< 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 >

```

Code example

```

package com . aliyun
import com . alibaba . fastjson . JSONArray ;
import com . alibaba . fastjson . JSONObject ;
import com . aliyuncs . DefaultAcs Client ;
import com . aliyuncs . exceptions . ClientException ;
import com . aliyuncs . mts . model . v20140618 . QueryJobLi
stRequest ;
import com . aliyuncs . mts . model . v20140618 . QueryJobLi
stResponse ;
import com . aliyuncs . mts . model . v20140618 . SubmitJobs
Request ;
import com . aliyuncs . mts . model . v20140618 . SubmitJobs
Response ;
import com . aliyuncs . profile . DefaultPro file ;
import org . apache . commons . codec . binary . Base64 ;
public class DataEncryp tionDemo {
    private DefaultAcs Client client = null ;

```

```

    private final String REGION = " cn - shanghai "; // set
as needed
    private final String MTS_ENDPOI NT = " mts . cn -
shanghai . aliyuncs . com "; // set as needed
    private final String ID =" idid "; // set as needed
    private final String KEY =" keykey "; // set as
needed
    private final String LOCATION = " oss - cn - shanghai
"; // set as needed
    private final String INPUT_BUCK ET = " input - bucket
"; // set as needed
    private final String OUTPUT_BUC KET = " output - bucket
"; // set as needed
    private final String PIPELINE_I D = " pipelineId "; //
set as needed
    public DataEncryp tionDemo () throws ClientExce ption {
        DefaultPro file . addEndpoin t ( REGION , REGION , " Mts
", MTS_ENDPOI NT );
        this . client = new DefaultAcs Client ( DefaultPro file
. getProfile ( REGION , ID , KEY ));
    }
    private JSONObject getInputFi le () {
        JSONObject inputFile = new JSONObject ();
        inputFile . put ( " Location ", LOCATION );
        inputFile . put ( " Bucket ", INPUT_BUCK ET );
        inputFile . put ( " Object ", " uploadvide o / test . flv ");
        return inputFile ;
    }
    private JSONArray getOutputs () {
        JSONArray outputs = new JSONArray ();
        outputs . add ( getOutput () );
        return outputs ;
    }
    private JSONObject getOutput () {
        JSONObject output = new JSONObject ();
        output . put ( " OutputObje ct ", " BaseTest / hls -
encryption . m3u8 ");
        output . put ( " TemplateId ", " S00000001 - 100020 ");
        output . put ( " Encryption ", getEncrypt ionConfigs ());
        return output ;
    }
    private JSONObject getEncrypt ionConfigs () {
        JSONObject encryption = new JSONObject ();
        encryption . put ( " Type ", " hls - aes - 128 ");
        encryption . put ( " Key ", Base64 . encodeBase 64URLSafeS
tring ( " encryption key123 ". getBytes ());
        encryption . put ( " KeyUri ", Base64 . encodeBase
64URLSafeS tring ( " http :// demo . aliyuncs . com / document /
hls128 . key ". getBytes ());
        encryption . put ( " KeyType ", " Base64 ");
        return encryption ;
    }
    private String submitJobs () throws ClientExce ption {
        JSONObject inputFile = getInputFi le ();
        SubmitJobs Request request = new SubmitJobs Request
();
        request . setInput ( inputFile . toJSONStri ng ());
        request . setOutputL ocation ( LOCATION );
        request . setOutputB ucket ( OUTPUT_BUC KET );
        request . setOutputs ( getOutputs (). toJSONStri ng ());
        request . setPipelin eId ( PIPELINE_I D );
        SubmitJobs Response reponse = this . client .
getAcsResp onse ( request );

```

```

        System.out.println ( JSONObject.toJSONString (
reponse.getJobResultList ());
        return reponse.getJobResultList ().get ( 0 ).getJob
().getJobId ();
    }
    public static void main ( String [] args ) throws
ClientException {
        DataEncryptionDemo demo = new DataEncryptionDemo
();
        String jobId = demo.submitJobs ();
    }
}

```

3.3.8 Create HLS standard encryption workflow

Overview

This document is an example of calling API to create HLS standard encryption workflow. For more information about creating workflow, see [AddMediaWorkflow](#).

Code dependency example

- For more information about MPS SDK, see [Installation](#).
- Other dependency.

```

< dependency >
  < groupId > com . alibaba </ groupId >
  < artifactId > fastjson </ artifactId >
  < version > 1 . 2 . 25 </ version >
</ dependency >

```

Code example

```

package com . aliyun . smallcode ;
import com . alibaba . fastjson . JSONArray ;
import com . alibaba . fastjson . JSONObject ;
import com . aliyuncs . DefaultAcs Client ;
import com . aliyuncs . exceptions . ClientException ;
import com . aliyuncs . mts . model . v20140618 . AddMediaWo
rkflowRequest ;
import com . aliyuncs . mts . model . v20140618 . AddMediaWo
rkflowResponse ;
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 "; //
Transcoding template ID , m3u8 template , set as
needed
    final String OSS_LOCATION = "< OssLocation >";
    final String INPUT_BUCKET = "< InputBucket >"; // Enter
bucket
    final String INPUT_PATH = "< InputPath >"; //如 " HLS -
Encryption "
    final String OUTPUT_BUCKET = "< OutputBucket >"; //
output bucket

```

```

final String ENCRYPTION_TYPE = " hls - aes - 128 ";
final String HLS_KEY_URI = "< Decryption key 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 encryption 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 , createTranscodeActivity ());
    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 createTranscodeActivity () {
    JSONObject transcodeActivity = new JSONObject ();
    transcodeActivity . put ( " Name " , ACT_ENCRYPTION );
    transcodeActivity . put ( " Type " , " Transcode " );
    transcodeActivity . put ( " Parameters " , buildTranscodeParameters ());
    return transcodeActivity ;
}
private JSONObject buildTranscodeParameters () {
    JSONObject transcodeParameters = new JSONObject ();

```

```

        transcodingParameters.put("OutputBucket", OUTPUT_BUCKET);
        transcodingParameters.put("OutputLocation", OSS_LOCATION);
        transcodingParameters.put("Outputs", buildOutputsConfig());
        return transcodingParameters;
    }

    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 subActivitiesOfStart = new JSONArray();
        subActivitiesOfStart.add(ACT_ENCRYPTION);
        dependencies.put(ACT_START, subActivitiesOfStart);
        JSONArray subActivitiesOfTranscode = new JSONArray();
        subActivitiesOfTranscode.add(ACT_REPORT);
        dependencies.put(ACT_ENCRYPTION, subActivitiesOfTranscode);
        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 MPS queues management

The system automatically creates an MPS queue when you open the MPS service.

You can also manage the MPS queue (pipeline) by using interfaces. For example:

SearchPipeline, QueryPipelineList, UpdatePipeline.

Search MPS queue

You can search the MPS queue information by using the SearchPipeline interface.

```
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 ();
// If an error occurs, it can throw ClientException or ServerException.
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 ());
}

```

Query MPS queue

If you know the pipelineId, you can use pipelineId to call QueryPipelineList interface to query MPS queue information.

```
String region = "< region >";
String accessKeyId = "< accessKeyId >";
String accessKeySecret = "< accessKeySecret >";
// known pipeline ID, seperated by comma (,)
String pipelineIds = "< pipelineIds >";
DefaultProfile profile = DefaultProfile.getProfile (
region, accessKeyId, accessKeySecret);
DefaultAcsClient client = new DefaultAcsClient (
profile);
QueryPipelineListRequest request = new QueryPipelineListRequest ();
request.setPipelineIds (pipelineIds);
// If an error occurs, it can throw ClientException or ServerException
QueryPipelineListResponse response = client.getAcsResponse (request);
List<QueryPipelineListResponse.Pipeline> pipelines = response.getPipelineList ();

```



```

        for ( QueryPipeLineListResponse pipeline : pipelines ) {
            System.out.println (" pipeline id : " + pipeline .
                getId () + ", name : " + pipeline . getName () + ", state : " +
                pipeline . getState ());
        }
    }

```

Update MPS queue

Use the `UpdatePipeline` interface to update MPS queue information, including MPS queue name and status. The status includes Active and Paused.

```

String region = "< region >";
String accessKeyId = "< accessKeyId >";
String accessKeySecret = "< accessKeySecret >";
DefaultProfile profile = DefaultProfile.getProfile (
    region, accessKeyId, accessKeySecret);
DefaultAcsClient client = new DefaultAcsClient (
    profile);
// Search user pipeline, the number is one by
// default.
SearchPipelineRequest searchPipelineRequest = new
SearchPipelineRequest ();
// If an error occurs, it can throw ClientException
// or ServerException.
SearchPipelineResponse searchPipelineResponse =
client.getAcsResponse ( searchPipelineRequest );
List< SearchPipelineResponse.Pipeline > pipelines =
searchPipelineResponse.getPipelineList ();
SearchPipelineResponse.Pipeline queryPipeline =
pipelines.get ( 0 );
// Update MPS queue status
UpdatePipelineRequest request = new UpdatePipelineRequest ();
request.setPipelineId ( queryPipeline.getId ());
request.setState ( " Paused " );
request.setName ( queryPipeline.getName ());
// If an error occurs, it can throw ClientException.
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 Query media - Use OSS file address

If you do not know the media ID (a live video converted to an on-demand video using the media workflow), you can use the media input URL to query the media information over `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 . QueryMedia
ListByURLR equest ;
import com . aliyuncs . mts . model . v20140618 . QueryMedia
ListByURLR esponse ;
import com . aliyuncs . profile . DefaultPro file ;
import java . io . Unsupporte dEncodingE xception ;
import java . net . URLEncoder ;
public class QueryMedia ListByURLD emo {
    private DefaultAcs Client client = null ;
    private final String REGION = "< region >";
    private final String ID ="< accessKeyI d >";
    private final String KEY ="< accessKeyS ecret >";
    public QueryMedia ListByURLD emo () throws ClientExce
ption {
        this . client = new DefaultAcs Client ( DefaultPro file
. getProfile ( REGION , ID , KEY ));
    }
    // Query the media informatio n , such as the media
ID , media status , and other attributes , based on
the OSS URL of the media source
    private void queryMedia ListByURL () throws ClientExce
ption , Unsupporte dEncodingE xception {
        QueryMedia ListByURLR equest request = new
QueryMedia ListByURLR equest ();
        String ossHost = ' http ://< input - bucket >.< region >
aliyuncs . com /';
        String ossObject = " test / The Legend of the
Swordsman . mp4 ";
        // ossObject must be RFC3986 - encoded
        String rfc3986Obj ect = encodeByRF C3986 ( ossObject );
        request . setFileURL s ( ossHost + rfc3986Obj ect );
        QueryMedia ListByURLR esponse response = this . client
. getAcsResp onse ( request );
        System . out . println ( JSONObject . toJSONStri ng (
response . getMediaLi st ());
    }
    private String encodeByRF C3986 ( String object ) throws
Unsupporte dEncodingE xception {
        StringBuil der builder = new StringBuil der ();
        String [] segments = object . split ("/");
        for ( int i = 0 ; i < segments . length ; i ++ ) {
            builder . append ( percentEnc ode ( segments [ i ]));
            if ( i != segments . length - 1 ) {
                builder . append ("/");
            }
        }
        return builder . toString ();
    }
    private static String percentEnc ode ( String value )
throws Unsupporte dEncodingE xception {
        if ( value == null )
            return null ;
        return URLEncoder . encode ( value , " UTF - 8 " ) . replace
("+", "% 20 " ) . replace ("*", "% 2A " ) . replace ("% 7E " , "~");
    }
    public static void main ( String [] args ) {
        try {
            QueryMedia ListByURLD emo demo = new QueryMedia
ListByURLD emo ();
            demo . queryMedia ListByURL ();
        } catch ( Exception e ) {
            e . printStack Trace ();
        }
    }
}

```

```
}

```

3.3.11 Add media

Add media file to Media Files, and the user can specify workflow ID to trigger the workflow, which then processes the video file:

```
package com . aliyun . mts ;
import com . alibaba . fastjson . JSONObject ;
import com . aliyuncs . DefaultAcs Client ;
import com . aliyuncs . exceptions . ClientExce ption ;
import com . aliyuncs . exceptions . ServerExce ption ;
import com . aliyuncs . mts . model . v20140618 . AddMediaRe
quest ;
import com . aliyuncs . mts . model . v20140618 . AddMediaRe
sponse ;
import com . aliyuncs . profile . DefaultPro file ;
import org . apache . commons . lang . exception . ExceptionU
tils ;
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 mtsEndpoin t = " mts ."
+ REGION + ". aliyuncs . com ";
    // Step 2 . set accesskey & keySecret
    private static String accessKeyI d = "";
    private static String accessKeyS ecret = "";
    private static DefaultAcs Client aliyunClie nt ;
    static {
        try {
            DefaultPro file . addEndpoin t ( REGION , REGION ,
" Mts " , mtsEndpoin t );
        } catch ( ClientExce ption e ) {
            System . out . print ( ExceptionU tils . getStackTr
ace ( e ));
            System . exit ( 1 );
        }
        aliyunClie nt = new DefaultAcs Client ( DefaultPro
file . getProfile ( REGION , accessKeyI d , accessKeyS ecret ));
    }
    public static void main ( String [] args ) throws
ClientExce ption {
        AddMediaRe quest request = new AddMediaRe quest
();
        request . setFileURL ( " http :// mtb - sz - in . oss - cn
- shenzhen . aliyuncs . com / media / r180 - ABC . mp4 " );
        request . setMediaWo rkflowId ( " 829bed0300 994057a49e
4f16de957e 34 " );
        try {
            AddMediaRe sponse response = aliyunClie nt .
getAcsResp onse ( request );
            System . out . println ( JSONObject . toJSONStri ng
( response ));
        } catch ( ServerExce ption e ) {
            System . out . println ( " Code :" + e . getErrCode
() + " Msg :" + e . getMessage ());
        } catch ( ClientExce ption e ) {

```

```

        System . out . println ( " Code :" + e . getErrCode
() + " Msg :" + e . getMessage ());
    }
}
}

```

3.3.12 Splicing and simple cutting

1. Create AcsClient instance.

```

DefaultPro file profile = DefaultPro file . getProfile (
    mpsRegionI d , // Region ID
    accessKeyI d , // AccessKey ID
    accessKeyS ecret ); // Access Key Secret
IAcsClient client = new DefaultAcs Client ( profile );

```

2. Create request and set parameters.

```

SubmitJobs Request request = new SubmitJobs Request ();

```

3. Set transcoding parameters.

· Input

```

JSONObject input = new JSONObject ();
input . put ( " Location ", ossLocatio n );
input . put ( " Bucket ", ossBucket );
try {
    input . put ( " Object ", URLEncoder . encode ( headObject
, " utf - 8 "));
} catch ( Unsupporte dEncodingException e ) {
    throw new RuntimeExc eption ( " input URL encode
failed " );
}
request . setInput ( input . toJSONStri ng ());

```

· Output

```

String outputOSSO bject ;
try {
    outputOSSO bject = URLEncoder . encode ( ossOutputO
bject , " utf - 8 " );
} catch ( Unsupporte dEncodingException e ) {
    throw new RuntimeExc eption ( " output URL encode
failed " );
}
JSONObject output = new JSONObject ();
output . put ( " OutputObje ct ", outputOSSO bject );
// Ouput -> TemplateId
output . put ( " TemplateId ", templateId );

```

- Video

```

JSONObject video = new JSONObject ();
video . put ( " Width ", " 1280 " );
video . put ( " Height ", " 720 " );

```

```
output . put ( " Video ", video . toJSONStri ng ());
```

- MergeList

```
JSONObject mergeVideo = new JSONObject ();
String mergeVideo URL ;
try {
    mergeVideo URL = String . format (
        " http ://% s .% s . aliyuncs .
com /% s ",
        ossBucket ,
        ossLocatio n ,
        URLEncoder . encode (
            ossInputOb ject , " utf - 8 "));
} catch ( UnsupportedEncodingE xception e ) {
    throw new RuntimeExc eption ( " mergeVideo URL
encode failed ");
}
mergeVideo . put ( " MergeURL ", mergeVideo URL );
JSONObject mergeTail = new JSONObject ();
String mergeTailU RL ;
try {
    mergeTailU RL = String . format (
        " http ://% s .% s . aliyuncs .
com /% s ",
        ossBucket ,
        ossLocatio n ,
        URLEncoder . encode (
            tailObject , " utf - 8 "));
} catch ( UnsupportedEncodingE xception e ) {
    throw new RuntimeExc eption ( " mergeTailU RL
encode failed ");
}
mergeTail . put ( " MergeURL ", mergeTailU RL );
JSONArray mergeList = new JSONArray ();
mergeList . add ( mergeVideo );
mergeList . add ( mergeTail );
output . put ( " MergeList ", mergeList . toJSONStri ng ());
```

4. Initiate API request and display returned value.

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

Full codes

```
package com . aliyun . mts ;
import java . io . UnsupportedEncodingE xception ;
import java . net . URLEncoder ;
import com . alibaba . fastjson . JSONArray ;
import com . alibaba . fastjson . JSONObject ;
```

```

import com . aliyuncs . profile . DefaultPro file ;
import com . aliyuncs . DefaultAcs Client ;
import com . aliyuncs . IAcsClient ;
import com . aliyuncs . exceptions . ClientExce ption ;
import com . aliyuncs . exceptions . ServerExce ption ;
import com . aliyuncs . mts . model . v20140618 .*;
public class Merge {
    private static String accessKeyI d = " xxx ";
    private static String accessKeyS ecret = " xxx ";
    private static String mpsRegionI d = " cn - hangzhou ";
    private static String pipelineId = " xxx ";
    private static String templateId = " S00000001 - 200030 ";
    private static String ossLocatio n = " oss - cn -
hangzhou ";
    private static String ossBucket = " xxx ";
    private static String ossInputOb ject = " input . mp4 ";
    private static String ossOutputO bject = " output . mp4
";
    private static String headObject = " head . mp4 ";
    private static String tailObject = " tail . mp4 ";
    public static void main ( String [] args ) {
        // DefaultAcs Client
        DefaultPro file profile = DefaultPro file . getProfile
(
            mpsRegionI d , // Region ID
            accessKeyI d , // AccessKey ID
            accessKeyS ecret ); // Access Key Secret
        IAcsClient client = new DefaultAcs Client ( profile );
        // request
        SubmitJobs Request request = new SubmitJobs Request
());
        // Input
        JSONObject input = new JSONObject ();
        input . put ( " Location " , ossLocatio n );
        input . put ( " Bucket " , ossBucket );
        try {
            input . put ( " Object " , URLEncoder . encode (
headObject , " utf - 8 "));
        } catch ( Unsupporte dEncodingE xception e ) {
            throw new RuntimeExc eption ( " input URL encode
failed " );
        }
        request . setInput ( input . toJSONStri ng ());
        // Output
        String outputOSSO bject ;
        try {
            outputOSSO bject = URLEncoder . encode ( ossOutputO
bject , " utf - 8 " );
        } catch ( Unsupporte dEncodingE xception e ) {
            throw new RuntimeExc eption ( " output URL encode
failed " );
        }
        JSONObject output = new JSONObject ();
        output . put ( " OutputObje ct " , outputOSSO bject );
        // Ouput -> TemplateId
        output . put ( " TemplateId " , templateId );
        // Ouput -> Video
        JSONObject video = new JSONObject ();
        video . put ( " Width " , " 1280 " );
        video . put ( " Height " , " 720 " );
        output . put ( " Video " , video . toJSONStri ng ());
        // Output -> MergeList
        JSONObject mergeVideo = new JSONObject ();
        String mergeVideo URL ;

```

```

        try {
            mergeVideo URL = String . format (
                " http ://% s .% s . aliyuncs
. com /% s ",
                ossBucket ,
                ossLocatio n ,
                URLEncoder . encode (
ossInputOb ject , " utf - 8 "));
        } catch ( Unsupporte dEncodingE xception e ) {
            throw new RuntimeExc eption (" mergeVideo URL
encode failed ");
        }
        mergeVideo . put (" MergeURL ", mergeVideo URL );
        JSONObject mergeTail = new JSONObject ();
        String mergeTailU RL ;
        try {
            mergeTailU RL = String . format (
                " http ://% s .% s . aliyuncs
. com /% s ",
                ossBucket ,
                ossLocatio n ,
                URLEncoder . encode (
tailObject , " utf - 8 "));
        } catch ( Unsupporte dEncodingE xception e ) {
            throw new RuntimeExc eption (" mergeTailU RL
encode failed ");
        }
        mergeTail . put (" MergeURL ", mergeTailU RL );
        JSONArray mergeList = new JSONArray ();
        mergeList . add ( mergeVideo );
        mergeList . add ( mergeTail );
        output . put (" MergeList ", mergeList . toJSONStri ng ());
        // Outputs
        JSONArray outputs = new JSONArray ();
        outputs . add ( output );
        request . setOutputs ( outputs . toJSONStri ng ());
        request . setOutputB ucket ( ossBucket );
        request . setOutputL ocation ( ossLocatio n );
        // PipelineId
        request . setPipelid ( pipelineId );
        // call api
        SubmitJobs Response response ;
        try {
            response = client . getAcsResp onse ( request );
            System . out . println (" RequestId is :"+ response .
getRequest
Id ());
            if ( response . getJobResu ltList (). get ( 0 ).
getSuccess () ) {
                System . out . println (" JobId is :"+ response .
getJobResu
ltList (). get ( 0 ). getJob (). getJobId ());
            } else {
                System . out . println (" SubmitJobs Failed code
:" + response . getJobResu ltList (). get ( 0 ). getCode () +
" message :"+ response .
getJobResu
ltList (). get ( 0 ). getMessage ());
            }
        } catch ( ServerExce ption e ) {
            e . printStack Trace ();
        } catch ( ClientExce ption e ) {
            e . printStack Trace ();
        }
    }
}

```

```
}

```

3.3.13 Add media workflow

The user can assemble activities provided by MPS, such as transcode activity and screenshot activity into a topology. The topology is as follows:

Topology type

```
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 = " Dependenci es " )
    private Map < String , List < String >> dependenci es ;
    public Map < String , List < String >> dependenci es () {
        return this . getDepende ncies () ;
    }
    public Topology () {
    }
    public Topology ( Map < String , Activity > activities , Map
< String , List < String >> dependenci es ) {
        this . setActivit ies ( activities ) ;
        this . setDepende ncies ( dependenci es ) ;
    }
    public Map < String , Activity > getActivit ies () {
        return activities ;
    }
    public Map < String , List < String >> getDepende ncies () {
        return dependenci es ;
    }
    public void setActivit ies ( Map < String , Activity >
activities ) {
        this . activities = activities ;
    }
    public void setDepende ncies ( Map < String , List < String
>> dependenci es ) {
        this . dependenci es = dependenci es ;
    }
}

```

ActivityType enumeration type

```
package com . aliyun . mts ;
/**
 * Created by zhongyizen gzy on 18 / 3 / 22 .
 */
public enum ActivityTy pe {
    Start , Transcode , Snapshot , MediaInfo , Analysis , Cover
, Summary , Censor , Report , UploadVeri fy , GenerateMa
sterPlayLi st , AudioGroup , SubtitleGr oup , PackageCon fig
}

```

Activity type

```
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 . getParamet ers () ;
    }
    public Activity ( String type , Map < String , String >
parameters ) {
        this . setType ( type ) ;
        this . setParamet ers ( parameters ) ;
    }
    public String getType () {
        return type ;
    }
    public Map < String , String > getParamet ers () {
        return parameters ;
    }
    public void setType ( String type ) {
        this . type = type ;
    }
    public void setParamet ers ( Map < String , String >
parameters ) {
        this . parameters = parameters ;
    }
}

```

AddMediaWorkflow type

```

package com . aliyun . mts ;
import com . alibaba . fastjson . JSONArray ;
import com . alibaba . fastjson . JSONObject ;
import com . aliyuncs . DefaultAcs Client ;
import com . aliyuncs . exceptions . ClientExce ption ;
import com . aliyuncs . exceptions . ServerExce ption ;
import com . aliyuncs . mts . model . v20140618 . AddMediaWo
rkflowRequ est ;
import com . aliyuncs . mts . model . v20140618 . AddMediaWo
rkflowResp onse ;
import com . aliyuncs . profile . DefaultPro file ;
import org . apache . commons . lang . exception . ExceptionU tils
;
import java . io . Unsupporte dEncodingE xception ;
import java . net . URLEncoder ;
import java . util . ArrayList ;
import java . util . Arrays ;
import java . util . HashMap ;
import java . util . List ;
public class AddMediaWo rkflow {
    // 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 mtsEndpoin t = " mts ." +
REGION + ". aliyuncs . com " ;
    // Step 2 . set accesskey & keySecret
    private static String accessKeyI d = "" ;

```

```

private static String accessKeySecret = "";
// Step 3 . set mps transcoding queue id
private static String PIPELINE_ID = "38bba54d52
4448be92d277caaa8da118";
private static DefaultAcsClient aliyunClient;
static {
    try {
        DefaultProfile.addEndpoint(REGION, REGION, "
Mts", mtsEndpoint);
    } catch (ClientException e) {
        System.out.println( "ExceptionUtils.getStackTrace
(e)");
        System.exit(1);
    }
    aliyunClient = new DefaultAcsClient( DefaultProfile
file.getProfile( REGION, accessKeyId, accessKeySecret ));
}
public static void main( String [] args ) throws
ClientException {
    AddMediaWorkflowRequest request = new AddMediaWo
rkflowRequest();
    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
.toString());
    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 . toString (
topology ));
        try {
            AddMediaWorkflowResponse response = aliyunClient
nt . getAcsResponse ( request );
            System . out . println ( JSONObject . toString (
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 Splicing-opening and ending scenes

1. Create AcsClient instance.

```

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

```

2. Create request and set parameters.

```

SubmitJobsRequest request = new SubmitJobsRequest ();

```

3. Set transcoding parameters.

· 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 . toJSONStri ng ());
```

• Output

```
String  outputOSSO bject ;
    try {
        outputOSSO bject = URLEncoder . encode ( ossOutputO
bject , " utf - 8 ");
    } catch ( Unsupporte dEncodingE xception e ) {
        throw new RuntimeExc eption (" output URL encode
failed ");
    }
    JSONObject  output = new  JSONObject ();
    output . put (" OutputObje ct ", outputOSSO bject );
    // Ouput -> TemplateId
    output . put (" TemplateId ", templateId );
```

- Video

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

- OpeningList

```
JSONObject  openingVid eo = new  JSONObject ();
String  openingVid eoURL ;
    try {
        openingVid eoURL = String . format (
            " http ://% s .% s . aliyuncs .
com /% s ",
            ossBucket ,
            ossLocatio n ,
            URLEncoder . encode (
                headObject , " utf - 8 "));
    } catch ( Unsupporte dEncodingE xception e ) {
        throw new RuntimeExc eption (" mergeVideo URL
encode failed ");
    }
    openingVid eo . put (" OpenUrl ", openingVid eoURL );
    openingVid eo . put (" Width ", " 640 ");
    openingVid eo . put (" Start ", " 2 ");
    JSONArray  openingVid eoList = new  JSONArray ();
    openingVid eoList . add ( openingVid eo );
    output . put (" OpeningLis t ", openingVid eoList .
toJSONStri ng ());
```

- TailSlateList

```
JSONObject  tailSlateV ideo = new  JSONObject ();
String  tailSlateV ideoURL ;
    try {
        tailSlateV ideoURL = String . format (
            " http ://% s .% s . aliyuncs .
com /% s ",
            ossBucket ,
            ossLocatio n ,
            URLEncoder . encode (
                tailObject , " utf - 8 "));
    } catch ( Unsupporte dEncodingE xception e ) {
```

```

        throw new RuntimeException (" mergeTailURL
        encode failed ");
    }
    tailSlateV ideo . put (" TailUrl ", tailSlateV ideoURL );
    tailSlateV ideo . put (" Width ", " 640 ");
    tailSlateV ideo . put (" BlendDuration ", " 3 ");
    tailSlateV ideo . put (" BgColor ", " Black ");
    JSONArray tailSlateV ideoList = new JSONArray ();
    tailSlateV ideoList . add ( tailSlateV ideo );
    output . put (" TailSlateList ", tailSlateV ideoList .
    toJSONString ());

```

4. Initiate API request and display returned value.

```

SubmitJobs Response response ;
response = client . getAcResp onse ( request );
System . out . println (" RequestId is :"+ response . getRequest
Id ());
if ( response . getJobResu ltList (). get ( 0 ). getSuccess () ) {
    System . out . println (" JobId is :"+ response .
getJobResu ltList (). get ( 0 ). getJob (). getJobId ());
} else {
    System . out . println (" SubmitJobs Failed code :"+
response . getJobResu ltList (). get ( 0 ). getCode () +
" message :"+ response . getJobResu
ltList (). get ( 0 ). getMessage ());
}

```

Full codes

```

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 . DefaultAcs Client ;
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 ) {
        // DefaultAcs Client
        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 . toString ());
    // Output
    String outputOSSObject ;
    try {
        outputOSSObject = URLEncoder . encode ( ossOutputO
bObject , " 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 -> 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 ( " mergeVideo URL
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 .
toString ());
    // 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 ");
    }
    tailSlateV ideo . put (" TailUrl ", tailSlateV ideoURL );
    tailSlateV ideo . put (" Width ", " 640 ");
    tailSlateV ideo . put (" BlendDurat ion ", " 3 ");
    tailSlateV ideo . put (" BgColor ", " Black ");
    JSONArray tailSlateV ideoList = new JSONArray ();
    tailSlateV ideoList . add ( tailSlateV ideo );
    output . put (" TailSlateL ist ", tailSlateV ideoList .
toJSONStri ng ());
    // Outputs
    JSONArray outputs = new JSONArray ();
    outputs . add ( output );
    request . setOutputs ( outputs . toJSONStri ng ());
    request . setOutputB ucket ( ossBucket );
    request . setOutputL ocation ( ossLocatio n );
    // PipelineId
    request . setPipelin eId ( pipelineId );
    // call api
    SubmitJobs Response response ;
    try {
        response = client . getAcSResp onse ( request );
        System . out . println (" RequestId is :"+ response .
getRequest Id ());
        if ( response . getJobResu ltList (). get ( 0 ).
getSuccess ()) {
            System . out . println (" JobId is :"+ response .
getJobResu ltList (). get ( 0 ). getJob (). getJobId ());
        } else {
            System . out . println (" SubmitJobs Failed code
:" + response . getJobResu ltList (). get ( 0 ). getCode () +
" message :"+ response .
getJobResu ltList (). get ( 0 ). getMessage ());
        }
    } catch ( ServerExce ption e ) {
        e . printStack Trace ();
    } catch ( ClientExce ption e ) {
        e . printStack Trace ();
    }
}
}
}

```

3.4 Python SDK

3.4.1 Overview

MPS Python SDK is based on Alibaba Cloud Python SDK, you must learn about the basic knowledge of Alibaba Cloud Python SDK.

- [Alibaba Cloud SDK Developer Guide > Python SDK > Quick start](#)
- [Alibaba Cloud SDK Developer Guide > Python SDK > User Guide > Use Python SDK](#)
- [Alibaba Cloud Python SDK GitHub](#)

After learning about the basic knowledge, you can perform MPS Python SDK installation. For more information, see [MPS > SDK Reference > Transcoding SDKs > Python SDK > Installation](#).

3.4.2 Installation

This article introduces pip installation method recommended by Alibaba Cloud Python SDK.

- Python 2.x

```

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

```

- Python 3.x

```

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

```

3.4.3 Quick start

1. Create AcsClient instance.

```

client = AcsClient ( access_key_id , access_key_secret ,
mps_region_id );

```

2. Create request and set parameters.

```

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

```

3. Initiate API request and display returned value.

```

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 ' ]

```

Full code

```

# -*- coding : utf8 -*-
import json
from aliyunsdkcore.client import AcsClient
from aliyunsdkmts.request.v20140618 import SearchPipe
lineRequest
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
    # Create AcsClient instance
    client = AcsClient ( access_key_id , access_key_secret ,
mps_region_id )
    # Create request, and set parameters
    request = SearchPipelineRequest.SearchPipelineRequest ()
    request.set_accept_format (' json ')
    # Initiate API request and display returned value
    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 Transcoding

1. Create AcsClient instance.

```

client = AcsClient ( access_key_id , access_key_secret ,
mps_region_id );

```

2. Create request, and set parameters.

```

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

```

3. Transcoding parameters.

• Input

```

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

```

• Output

```

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

```

- Container

```

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

```

- Video

```

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

```

```
' Fps ': 25 }
```

- Audio

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

- TemplateId

```
output [' TemplateId '] = template_i d
```

· PipelineId

```
request . set_Pipeli neId ( pipeline_i d )
```

4. Initiate API request and display returned value.

```
response_s tr = client . do_action_ with_excep tion ( request
)
response = json . loads ( response_s tr )
print ' RequestId is :', response [' RequestId ']
if response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Success
']:
print ' JobId is :', response [' JobResultL ist '][ ' JobResult
'][ 0 ][ ' Job '][ ' JobId ']
else :
print ( ' SubmitJobs Failed code :',
response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Code '],
' message :',
response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Message
'])
```

Full code

```
# -*- coding : utf8 -*-
import json
from urllib import quote
from aliynsdkc ore . client import AcsClient
from aliynsdkm ts . request . v20140618 import SubmitJobs
Request
access_key _id = ' xxx '
access_key _secret = ' xxx '
mps_region _id = ' cn - hangzhou '
pipeline_i d = ' xxx '
template_i d = ' S00000001 - 200010 '
oss_locati on = ' oss - cn - hangzhou '
oss_bucket = ' xxx '
oss_input_ object = ' input . mp4 '
oss_output_ object = ' output . mp4 '
Create AcsClient instance
client = AcsClient ( access_key _id , access_key _secret ,
mps_region _id );
Create request, and set parameters
request = SubmitJobs Request . SubmitJobs Request ()
request . set_accept _format ( ' json ' )
# Input
job_input = { ' Location ': oss_locati on ,
' Bucket ': oss_bucket ,
```

```

        ' Object ': quote ( oss_input_ object ) }
    request . set_Input ( json . dumps ( job_input ))
# Output
    output = {' OutputObje ct ': 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_i d
    outputs = [ output ]
    request . set_Output s ( json . dumps ( outputs ))
    request . set_Output Bucket ( oss_bucket )
    request . set_Output Location ( oss_locati on )
# PipelineId
    request . set_Pipeli neId ( pipeline_i d )
    Initiate API request and diaplay returned value
    response_s tr = client . do_action_ with_excep tion ( request )
    response = json . loads ( response_s tr )
    print ' RequestId is :', response [' RequestId ' ]
    if response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Success ']:
        print ' JobId is :', response [' JobResultL ist '][ '
JobResult '][ 0 ][ ' Job '][ ' JobId ' ]
    else :
        print ( ' SubmitJobs Failed code :',
                response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Code
'],
                ' message :',
                response [' JobResultL ist '][ ' JobResult '][ 0 ][ '
Message ' ])

```

3.4.5 Watermarks

1. Create AcsClient instance.

```

client = AcsClient ( access_key _id , access_key _secret ,
mps_region _id );

```

2. Create request, and set parameters.

```

request = SubmitJobs Request . SubmitJobs Request ()
request . set_accept _format ( ' json ' )

```

3. Watermark parameters.

- Image watermark

```

image_wate rmark_inpu t = {
    ' Location ': oss_locati on ,
    ' Bucket ': oss_bucket ,
    ' Object ': quote ( image_wate rmark_obje
ct )
}

```

```

    }
    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 + V5paH5a2X5 rC05Y2w',
  '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
}

```

4. Initiate API request and display returned value.

```

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 '])

```

Full codes

```

# -*- coding : utf8 -*-
from pprint import pprint
import json
from urllib import quote
from aliyunsdkc ore . client import AcsClient
from aliyunsdkm ts . request . v20140618 import SubmitJobs
Request
access_key _id = ' xxx '
access_key _secret = ' xxx '
mps_region _id = ' cn - hangzhou '
pipeline_i d = ' xxx '
watermark_ template_i d = ' xxx '
template_i d = ' S00000001 - 200030 '
oss_locati on = ' oss - cn - hangzhou '
oss_bucket = ' xxx '
oss_input_ object = ' input . mp4 '
oss_output _object = ' output . mp4 '
image_wate rmark_obje ct = ' logo . png '
video_wate rmark_obje ct = ' logo . mov '
# AcsClient
client = AcsClient ( access_key _id , access_key _secret ,
mps_region _id );
# request
request = SubmitJobs Request . SubmitJobs Request ()
request . set_accept _format ( ' json ' )
# Input
job_input = { ' Location ': oss_locati on ,
' Bucket ': oss_bucket ,
' Object ': quote ( oss_input_ object ) }
request . set_input ( json . dumps ( job_input ))
# Output
output = { ' OutputObje ct ': quote ( oss_output _object ) }
# Ouput -> TemplateId
output [ ' TemplateId ' ] = template_i d
## Image Watermark
image_wate rmark_inpu t = { ' Location ': oss_locati on ,
' Bucket ': oss_bucket ,
' Object ': quote ( image_wate rmark_obje
ct ) }
image_wate rmark = {
' WaterMarkT emplateId ': watermark_ template_i
d ,
' Type ': ' Image ',
' InputFile ': image_wate rmark_inpu t ,
' ReferPos ': ' TopRight ',
' Width ': 0 . 05 ,
' Dx ': 0 ,
' Dy ': 0
}
## Text Watermark
text_config = {
' Content ': ' 5rWL6K + V5paH5a2X5 rC05Y2w ',
' FontName ': ' SimSun ',
' FontSize ': 16 ,
' FontColor ': ' Red ',

```

```

        ' FontAlpha ': 0.5 ,
        ' Top ': 10 ,
        ' Left ': 10
    }
    text_water mark = {
        ' WaterMarkT emplateId ': watermark_ template_i
d ,
        ' Type ': ' Text ',
        ' TextWaterM ark ': text_confi g
    }
## Video Watermark
video_wate rmark_inpu t = {' Location ': oss_locati on ,
    ' Bucket ': oss_bucket ,
    ' Object ': quote ( video_wate rmark_obje
ct ) }
video_wate rmark = {
    ' WaterMarkT emplateId ': watermark_ template_i
d ,
    ' Type ': ' Image ',
    ' InputFile ': video_wate rmark_inpu t ,
    ' ReferPos ': ' BottomLeft ',
    ' Height ': 240 ,
    ' Dx ': 0 ,
    ' Dy ': 0
}
# Output -> Watermarks
watermarks = [ image_wate rmark , text_water mark , video_wate
rmark ]
output [' WaterMarks '] = watermarks
# Outputs
outputs = [ output ]
request . set_Output s ( json . dumps ( outputs ))
request . set_Output Bucket ( oss_bucket )
request . set_Output Location ( oss_locati on )
# PipelineId
request . set_Pipeli neId ( pipeline_i d )
# call api
response_s tr = client . do_action_ with_excep tion ( request )
response = json . loads ( response_s tr )
print ' RequestId is :', response [' RequestId ']
if response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Success ']:
    print ' JobId is :', response [' JobResultL ist '][ '
JobResult '][ 0 ][ ' Job '][ ' JobId ' ]
else :
    print (' SubmitJobs Failed code :',
        response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Code
'],
        ' message :',

```

```
response [' JobResultL ist '][' JobResult '][ 0 ]['
Message '])
```

3.4.6 Screenshot

1. Create an AcsClient instance.

```
client = AcsClient ( access_key _id , access_key _secret ,
mps_region _id );
```

2. Create request and set parameters.

```
request = SubmitSnapshotJobRequest . SubmitSnapshotJobRequest (
request . set_accept _format ( ' json ' )
```

3. Set screenshot parameters.

- **Input**

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

- **SnapshotConfig**

- **OutputFile**

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

- **Time**

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

- **Interval/Num**

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

- **Width/Height**

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

4. Initiate API request and display returned value.

```
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_000 01 . jpg '
% ( oss_bucket , oss_location )
```

```
print ' http ://% s .% s . aliyuncs . com / output_000 02 . jpg '
% ( oss_bucket , oss_locati on )
print ' http ://% s .% s . aliyuncs . com / output_000 03 . jpg '
% ( oss_bucket , oss_locati on )
```

Complete code

```
# -*- coding : utf8 -*-
import json
from urllib import quote
from aliyunsdkc ore . client import AcsClient
from aliyunsdkm ts . request . v20140618 import SubmitSnap
shotJobReq uest
access_key _id = ' xxx '
access_key _secret = ' xxx '
mps_region _id = ' cn - hangzhou '
pipeline_i d = ' xxx '
oss_locati on = ' 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 = SubmitSnap shotJobReq uest . SubmitSnap shotJobReq
uest ()
request . set_accept _format ( ' json ' )
# Input
job_input = { ' Location ' : oss_locati on ,
' Bucket ' : oss_bucket ,
' Object ' : quote ( oss_input_ object ) }
request . set_Input ( json . dumps ( job_input ) )
# SnapshotCo nfig -> OutputFile
job_output = { ' Location ' : oss_locati on ,
' Bucket ' : oss_bucket ,
' Object ' : quote ( oss_output _object ) }
snapshot_c onfig = { ' OutputFile ' : job_output }
# SnapshotCo nfig -> Time
snapshot_c onfig [ ' Time ' ] = 2
# SnapshotCo nfig -> Interval / Num
snapshot_c onfig [ ' Interval ' ] = 2
snapshot_c onfig [ ' Num ' ] = 3
# SnapshotCo nfig -> Width / Height
snapshot_c onfig [ ' Height ' ] = 360
# SnapshotCo nfig
request . set_Snapsh otConfig ( json . dumps ( snapshot_c onfig ) )
# PipelineId
request . set_Pipeli neId ( pipeline_i d )
# call api
response_s tr = client . do_action_ with_excep tion ( request )
response = json . loads ( response_s tr )
print response
print ' RequestId is : ' , response [ ' RequestId ' ]
print ' JobId is : ' , response [ ' SnapshotJo b ' ] [ ' Id ' ]
print ' http ://% s .% s . aliyuncs . com / output_000 01 . jpg ' %
( oss_bucket , oss_locati on )
print ' http ://% s .% s . aliyuncs . com / output_000 02 . jpg ' %
( oss_bucket , oss_locati on )
```



```
print ' http ://% s .% s . aliyuncs . com / output_000 03 . jpg ' %
( oss_bucket , oss_locati on )
```

3.4.7 Query media - Use OSS file address

If you do not know the media ID (a live video converted to an on-demand video using the media workflow), you can use the media input URL to query the media information over `QueryMediaListByURL`.

```
import json
from aliyunsdkm ts . request . v20140618 import QueryMedia
ListByURLR equest
from aliyunsdkc ore import client
import urllib
region = '< region >'
access_key _id = '< accessKeyI d >'
access_key _secret = '< accessKeyS ecret >'
def queryMedia ListByURL ():
    global client
    client = client . AcsClient ( access_key _id , access_key
    _secret , region )
    request = QueryMedia ListByURLR equest . QueryMedia
    ListByURLR equest ()
    ossDomain = ' http ://< input - bucket >.< region > . aliyuncs .
    com /';
    # Encode ossObject
    ossObject = encodeByRF C3986 (" test / The Legend of the
    Swordsman . mp4 ")
    request . set_FileUR Ls ( ossDomain + ossObject )
    response = client . do_action_ with_excep tion ( request );
    json_respo nse = json . loads ( response )
    print json_respo nse
def encodeByRF C3986 ( ossObject ):
    return urllib . quote ( ossObject )
if __name__ == " __main__ ":
    queryMedia ListByURL ()
```

3.4.8 Create HLS standard encryption workflow

Introduction

This document is an example of calling API to create HLS standard encryption workflow. For more information about creating HLS standard encryption workflow and playing encryption videos, see [HLS encryption and play](#). For more information about MPS SDK, see [Installation](#).

Example

```
import json
from aliyunsdkm ts . request . v20140618 import AddMediaWo
rkflowRequ est
from aliyunsdkc ore import client
REGION_ID = '< region >'
ACCESS_KEY _ID = '< accessKeyI d >'
ACCESS_KEY _SECRET = '< accessKeyS ecret >'
```

```

PIPELINE_ID = "< PipelineId >"
TEMPLATE_ID = "S00000001 - 100020 " # Transcoding template
ID, m3u8 template, set as needed
OSS_LOCATION = "< OssLocation >"
INPUT_BUCKET = "< InputBucket >"
INPUT_PATH = "< InputPath >" ## Example : " HLS - Encryption "
OUTPUT_BUCKET_KEY = "< OutputBucket >"
ENCRYPTION_TYPE = "hls - aes - 128 "
HLS_KEY_URI = "< URI of decryption key >" # Example : 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.AddMediaWorkflowRequ
est()
    request.set_Topology(buildWorkflowTopology())
    request.set_Name("HLS encryption workflow.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"] = buildTranscodeParame
ters()
    return transcodeActivity
def buildTranscodeParameters():
    transcodeParameters = {}
    transcodeParameters["OutputBucket"] = OUTPUT_BUCKET
    transcodeParameters["OutputLocation"] = OSS_LOCATION
    transcodeParameters["Outputs"] = buildOutputsConfig()

```

```

return transcodeParameters
def buildOutput tsConfig():
    outputs = []
    output = {}
    output [" ObjectRegeneration " ] = 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 Splicing and simple cutting

1. Create AcsClient instance.

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

2. Create request and set parameters.

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

3. Sets transcode parameters.

· Input

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

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

- **Output**

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

- **Video**

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

- **MergeList**

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

4. Initiate API request and display returned value.

```
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'])
```

Full codes

```
# -*- 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 = SubmitJobs Request . SubmitJobs Request ()
request . set_accept _format (' json ')
# Input
job_input = {' Location ': oss_locati on ,
              ' Bucket ': oss_bucket ,
              ' Object ': quote ( head_objec t ) }
request . set_Input ( json . dumps ( job_input ))
# Output
output = {' OutputObje ct ': quote ( oss_output _object )}
# Output -> TemplateId
output [' TemplateId '] = template_i d
# Output -> Video
output [' Video '] = {' Width ': 1280 ,
                     ' Height ': 720 }
# Output -> MergeList
merge_vide o = {' MergeURL ': ' http ://% s .% s . aliyuncs . com /
% s '( oss_bucket , oss_locati on , quote ( oss_input_ object ))}
merge_tail = {' MergeURL ': ' http ://% s .% s . aliyuncs . com /% s
' %( oss_bucket , oss_locati on , quote ( tail_objec t ))}
output [' MergeList '] = [ merge_vide o , merge_tail ]
# Outputs
outputs = [ output ]
request . set_Output s ( json . dumps ( outputs ))
request . set_Output Bucket ( oss_bucket )
request . set_Output Location ( oss_locati on )
# PipelineId
request . set_Pipeli neId ( pipeline_i d )
# call api
response_s tr = client . do_action_ with_excep tion ( request )
response = json . loads ( response_s tr )
print ' RequestId is :', response [' RequestId ']
if response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Success ']:
    print ' JobId is :', response [' JobResultL ist '][ '
JobResult '][ 0 ][ ' Job '][ ' JobId ']
else :
    print (' SubmitJobs Failed code :',
          response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Code
'],
          ' message :',
          response [' JobResultL ist '][ ' JobResult '][ 0 ][ '
Message '])

```

3.4.10 Splicing-opening and ending scenes

1. Create AcsClient instance.

```

client = AcsClient ( access_key _id , access_key _secret ,
mps_region _id );

```

2. Create request and set parameters.

```

request = SubmitJobs Request . SubmitJobs Request ()

```

```
request . set_accept _format ( ' json ' )
```

3. Sets transcode parameters.

- **Input**

```
job_input = { ' Location ': oss_locati on ,
              ' Bucket ': oss_bucket ,
              ' Object ': quote ( head_objec t ) }
request . set_Input ( json . dumps ( job_input ) )
```

- **Output**

```
output = { ' OutputObje ct ': quote ( oss_output _object ) }
```

- **Video**

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

- **OpeningList**

```
opening_vi deo = { ' OpenUrl ': ' http ://% s .% s . aliyuncs
                  . com /% s '%( oss_bucket , oss_locati on , quote (
                  head_objec t ) ) ,
                  ' Width ': 640 ,
                  ' Start ': 2 }
output [ ' OpeningLis t ' ] = [ opening_vi deo ]
```

- **TailSlateList**

```
tailslate_ video = { ' TailUrl ': ' http ://% s .% s .
                    aliyuncs . com /% s '%( oss_bucket , oss_locati on , quote
                    ( tail_objec t ) ) ,
                    ' Width ': 640 ,
                    ' BlendDurat ion ': 3 ,
                    ' BgColor ': ' Black ' }
output [ ' TailSlateL ist ' ] = [ tailslate_ video ]
```

4. Initiate API request and display returned value.

```
response_s tr = client . do_action_ with_excep tion ( request
)
response = json . loads ( response_s tr )
print ' RequestId is :', response [ ' RequestId ' ]
if response [ ' JobResultL ist ' ] [ ' JobResult ' ] [ 0 ] [ ' Success
' ] :
    print ' JobId is :', response [ ' JobResultL ist ' ] [ '
JobResult ' ] [ 0 ] [ ' Job ' ] [ ' JobId ' ]
else :
    print ( ' SubmitJobs Failed code :',
          response [ ' JobResultL ist ' ] [ ' JobResult ' ] [ 0 ] [ '
Code ' ],
          ' message :',
```

```
response [' JobResultL ist '][' JobResult '][ 0 ]['
Message '])
```

Full codes

```
# -*- coding : utf8 -*-
import json
from urllib import quote
from aliyunsdkc ore . client import AcsClient
from aliyunsdkm ts . request . v20140618 import SubmitJobs
Request
access_key _id = ' xxx '
access_key _secret = ' xxx '
mps_region _id = ' cn - hangzhou '
pipeline_i d = ' xxx '
template_i d = ' S00000001 - 200030 '
oss_locati on = ' oss - cn - hangzhou '
oss_bucket = ' xxx '
oss_input_ object = ' input . mp4 '
oss_output _object = ' output . mp4 '
head_objec t = ' head . mp4 '
tail_objec t = ' tail . mp4 '
# AcsClient
client = AcsClient ( access_key _id , access_key _secret ,
mps_region _id );
# request
request = SubmitJobs Request . SubmitJobs Request ()
request . set_accept _format ( ' json ' )
# Input
job_input = { ' Location ': oss_locati on ,
' Bucket ': oss_bucket ,
' Object ': quote ( oss_input_ object ) }
request . set_Input ( json . dumps ( job_input ) )
# Output
output = { ' OutputObje ct ': quote ( oss_output _object ) }
# Ouput -> TemplateId
output [ ' TemplateId ' ] = template_i d
# Output -> OpeningLis t
opening_vi deo = { ' OpenUrl ': ' http ://% s .% s . aliyuncs . com
/% s '%( oss_bucket , oss_locati on , quote ( head_objec t ) ) ,
' Width ': 640 ,
' Start ': 2 }
output [ ' OpeningLis t ' ] = [ opening_vi deo ]
# Output -> TailSlateL ist
tailslate_ video = { ' TailUrl ': ' http ://% s .% s . aliyuncs .
com /% s '%( oss_bucket , oss_locati on , quote ( tail_objec t
) ) ,
' Width ': 640 ,
' BlendDurat ion ': 3 ,
' BgColor ': ' Black ' }
output [ ' TailSlateL ist ' ] = [ tailslate_ video ]
# Outputs
outputs = [ output ]
request . set_Output s ( json . dumps ( outputs ) )
request . set_Output Bucket ( oss_bucket )
request . set_Output Location ( oss_locati on )
# PipelineId
request . set_Pipeli neId ( pipeline_i d )
# call api
response_s tr = client . do_action_ with_excep tion ( request )
response = json . loads ( response_s tr )
print ' RequestId is :', response [ ' RequestId ' ]
if response [ ' JobResultL ist '][' JobResult '][ 0 ][' Success ']:
```

```

    print ' JobId is :', response [' JobResultL ist '][ '
JobResult '][ 0 ][ ' Job '][ ' JobId '
else :
    print (' SubmitJobs Failed code :',
          response [' JobResultL ist '][ ' JobResult '][ 0 ][ ' Code
'],
          ' message :',
          response [' JobResultL ist '][ ' JobResult '][ 0 ][ '
Message '])

```

3.4.11 Add media

Add a video file to MEdia Files, and the user can specify workflow to be triggered to process this file:

```

import json
from aliyunsdkc ore . acs_except ion . exceptions import
ServerExce ption , ClientExce ption
from aliyunsdkm ts . request . v20140618 import AddMediaRe
quest
from aliyunsdkc ore import client
import urllib
import thread
# Step 1 set region
REGION = " cn - shenzhen ";
mtsEndpoin t = " mts ." + REGION + ". aliyuncs . com ";
# Step 2 . set accesskey & keySecret
accessKeyI d = "";
accessKeyS ecret = "";
cli = client . AcClient ( accessKeyI d , accessKeyS ecret ,
REGION )
def addMeida ():
    request = AddMediaRe quest . AddMediaRe quest ()
    request . set_FileUR L (" http :// mtb - sz - in . oss - cn -
shenzhen . aliyuncs . com / media / r180 - ABC . mp4 ")
    request . set_MediaW orkflowId (" 829bed0300 994057a49e
4f16de957e 34 ")
    try :
        response = cli . do_action_ with_excep tion ( request )
        json_respo nse = json . loads ( response )
        print json . dumps ( json_respo nse )
    except ServerExce ption , e :
        print e . get_error_ code (), e . get_error_ msg ()
    except ClientExce ption , e :
        print e . get_error_ code (), e . get_error_ msg ()
def encodeByRF C3986 ( ossObject ):
    return urllib . quote ( ossObject )
if __name__ == " __main__ ":
    addMeida ()

```

3.4.12 Add media workflow

The user can assemble activities provided by MPS, such as transcode activity and screenshot activity into a topology. The topology is as follows:

```

import json
from aliyunsdkc ore . acs_except ion . exceptions import
ServerExce ption

```



```

from aliyunsdkm ts . request . v20140618 import AddMediaWo
rkflowRequ est
from aliyunsdkc ore import client
import urllib
import thread
# Step 1 set region
REGION = " cn - shenzhen ";
OSS_REGION = " oss - cn - shenzhen ";
mtsEndpoin t = " mts ." + REGION + ". aliyuncs . com ";
# Step 2 . set accesskey & keySecret
accessKeyI d = "";
accessKeyS ecret = "";
# Step 3 . set mps transcodin g queue id
PIPELINE_I D = " 38bba54d52 4448be92d2 77caaa8da1 18 ";
cl = client . AcsClient ( accessKeyI d , accessKeyS ecret ,
REGION )
def addMeidaWo rkflow () :
request = AddMediaWo rkflowRequ est . AddMediaWo rkflowRequ
est ()
request . set_Name ( " Sequential - workflow " );
startActiv ity = {
" Type " : " Start " ,
" Parameters " : {
" InputFile " : {
" Bucket " : " mtb - sz - in " ,
" Location " : OSS_REGION ,
" ObjectPref ix " : " media /"
},
" PipelineId " : PIPELINE_I D
}
}
transcodeA ctivity = {
" Type " : " Transcode " ,
" Parameters " : {
" Outputs " : [
{
" OutputObje ct " : encodeByRF C3986 ( "
transcode /{ ObjectPref ix }/{ FileName }.{ ExtName }" ) ,
" TemplateId " : " S00000001 - 000070 "
}
],
" OutputLoca tion " : OSS_REGION ,
" OutputBuck et " : " mtb - sz - out "
}
}
reportActi vity = {
" Type " : " Report " ,
" Parameters " : {
}
}
}
topology = {
" Activities " : {
" startNode " : startActiv ity ,
" transcodin gNode " : transcodeA ctivity ,
" reportNode " : reportActi vity
},
" Dependenci es " : {
" startNode " : [ " transcodin gNode " ] ,
" transcodin gNode " : [ " reportNode " ] ,
" reportNode " : [ ]
}
}
}
request . set_Topolo gy ( topology )
try :

```

```
        response = json . loads ( cl . do_action_ with_excep tion
( request ))
        print json . dumps ( response )
    except ServerExce ption , e :
        print e . get_error_ code ( ) , e . get_error_ msg ( )
def encodeByRF C3986 ( ossObject ) :
    return urllib . quote ( ossObject )
if __name__ == " __main__ " :
    addMeidaWo rkflow ( )
```

3.5 PHP SDK

3.5.1 Overview

MPS PHP SDK is based on Alibab Cloud PHP SDK. This article introduces the basic knowledge about Alibaba Cloud PHP SDK.

- [Alibaba Cloud SDK Developer Guide > PHP SDK > User Guide](#)
- [Alibaba Cloud PHP SDK GitHub](#)

After learning about the basic knowledge, you can perform MPS PHP SDK installation. For more information, see [MPS > SDK Reference > Transcoding SDKs > PHP SDK > Installation](#).

3.5.2 Installation

This article introduces the installation method of Alibaba Cloud PHP SDK.

1. Download the source code.

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

2. Add the reference.

Assume the PHP SDK is downloaded to the path `/path/to/aliyun-openapi-php-sdk`.

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

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

Assume `aliyun-openapi-php-sdk` is referred to under the current directory of the project.

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

3. Automatically load MPS SDK.

Edit the file `aliyun-openapi-php-sdk/aliyun-php-sdk-core/Config.php`.

Find the content `// config_sdk_auto_load_path`, and add

```
Autoloader::addAutoloadPath("aliyun-php-sdk-mts");
```

at the end.

3.5.3 Quick start

1. Create AcsClient instance.

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

2. Create request and set parameters.

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

3. Initiate API request and display returned value.

```
$response = $client->getAcsResponse($request);
```

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

Full codes

```
<? 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 " ;
    # Create DefaultAcs Client instance and complete
    initialization
    $ clientProfile = DefaultProfile :: getProfile (
        $ mps_region _id ,           # Your Region ID
        $ access_key _id ,           # Your AccessKey
        ID
        $ access_key _secret         # Your AccessKey
        Secret
    );
    $ client = new DefaultAcs Client ( $ clientProfile );
    # Create API request and set parameters
    $ request = new Mts \ SearchPipelineRequest ();
    # Initiate request and handle the returned value
    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 Transcoding

1. Create AcsClient instance.

```
$ clientProfile = DefaultProfile :: getProfile (
$ mps_region _id ,           # Your Region ID
$ access_key _id ,           # Your AccessKey ID
$ access_key _secret         # Your AccessKey Secret
);
```

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

2. Create request, and set parameters.

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

3. Transcoding parameters.

• Input

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

• Output

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

- Container

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

- Video

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

- Audio

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

- TemplateId

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

• PipelineId

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

4. Initiate API request and display returned value.

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

```

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

```

Full code

```

<? php
include_on ce ' 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_i d = ' xxx ' ;
$ template_i d = ' S00000001 - 200010 ' ;
$ oss_locati on = ' oss - cn - hangzhou ' ;
$ oss_bucket = ' xxx ' ;
$ oss_input_ object = ' input . mp4 ' ;
$ oss_output _object = ' output . mp4 ' ;
# Create DefaultAcs Client instance and complete
initializa tion
$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id , # Your Region ID
    $ access_key _id , # Your AccessKey ID
    $ access_key _secret # Your AccessKey
    Secret
);
$ client = new DefaultAcs Client ($ clientProf ile );
# Create API request and set parameters
$ request = new Mts \ SubmitJobs Request ();
$ request -> setAcceptF ormat (' JSON ');
# Input
$ input = array (' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ oss_input_ object ));
$ request -> setInput ( json_encod e ($ input ));
# Output
$ output = array (' OutputObje ct ' => urlencode ($ oss_output
_object ));
# Ouput -> Container
$ output [' Container '] = array (' Format ' => ' mp4 ');
# Ouput -> Video
$ output [' Video '] = array (' Codec ' => ' H . 264 ',
    ' Bitrate ' => 1500 ,
    ' Width ' => 1280 ,
    ' Fps ' => 25 );
# Ouput -> Audio
$ output [' Audio '] = array (' Codec ' => ' AAC ',
    ' Bitrate ' => 128 ,
    ' Channels ' => 2 ,
    ' Samplerate ' => 44100 );
# Ouput -> TemplateId
$ output [' TemplateId '] = $ template_i d ;
$ outputs = array ($ output );
$ request -> setOuputs ( json_encod e ($ outputs ));
$ request -> setOutputB ucket ($ oss_bucket );
$ request -> setOutputL ocation ($ oss_locati on );
# PipelineId
$ request -> setPipelid ($ pipeline_i d );

```

```
# Initiate request and handle returned value
try {
    $ response = $ client -> getAcResp onse ($ request );
    print ' RequestId is : ' . $ response ->{' RequestId ' } . "\ n
";
    if ($ response ->{' JobResultL ist '}->{' JobResult '}[ 0 ]->{'
Success '}) {
        print ' JobId is : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Job '}->{' JobId ' } . "\ n ";
    } else {
        print ' SubmitJobs Failed code : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Code ' } .
            message : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Message ' } . "\ n ";
    }
} catch ( ServerExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
} catch ( ClientExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
}
```

3.5.5 Watermarks

1. Create AcClient instance.

```
$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id , # Region ID
    $ access_key _id , # AccessKey ID
    $ access_key _secret # AccessKey Secret
);
$ client = new DefaultAc Client ($ clientProf ile );
```

2. Create request and set parameters.

```
$ request = new Mts \ SubmitJobs Request ();
$ request -> setAcceptF ormat ( ' JSON ' );
```

3. Watermark parameters.

• Image watermark

```
$ image_wate rmark_inpu t = array (
    ' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ image_wate
rmark_obje ct )
);
$ image_wate rmark = array (
    ' WaterMarkT emplateId ' => $ watermark_
template_i d ,
    ' Type ' => ' Image ',
    ' InputFile ' => $ image_wate rmark_inpu t ,
    ' ReferPos ' => ' TopRight ',
    ' Width ' => 0 . 05 ,
    ' Dx ' => 0 ,
```

```
        ' Dy ' => 0
    );
```

- **Text watermark**

```
$ text_configng = array (
    ' Content ' => ' 5rWL6K + V5paH5a2X5 rC05Y2w ',
    ' FontName ' => ' SimSun ',
    ' FontSize ' => 16 ,
    ' FontColor ' => ' Red ',
    ' FontAlpha ' => 0 . 5 ,
    ' Top ' => 10 ,
    ' Left ' => 10
);
$text_watermark = array (
    ' WaterMarkTemplateId ' => $ watermark_
template_id ,
    ' Type ' => ' Text ',
    ' TextWatermark ' => $ text_configng
);
```

- **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
);
```

4. Initiate API request and display returned value.

```
$ 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";
}
```



```
}

```

Full codes

```
<? 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_i d = ' xxx ' ;
$ watermark_ template_i d = ' xxx ' ;
$ template_i d = ' S00000001 - 200030 ' ;
$ oss_locati on = ' oss - cn - hangzhou ' ;
$ oss_bucket = ' presigned ' ;
$ oss_input_ object = ' input . mp4 ' ;
$ oss_output _object = ' output . mp4 ' ;
$ image_wate rmark_obje ct = ' logo . png ' ;
$ video_wate rmark_obje ct = ' logo . mov ' ;
# DefaultAcs Client
$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id , # Region ID
    $ access_key _id , # AccessKey ID
    $ access_key _secret # AccessKey Secret
);
$ client = new DefaultAcs Client ($ clientProf ile );
# request
$ request = new Mts \ SubmitJobs Request ();
$ request -> setAcceptF ormat (' JSON ');
# Input
$ input = array (' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ oss_input_ object ));
$ request -> setInput ( json_encod e ($ input ));
# Output
$ output = array (' OutputObje ct ' => urlencode ($ oss_output
_object ));
# Output -> TemplateId
$ output [' TemplateId '] = $ template_i d ;
## Image Watermark
$ image_wate rmark_inpu t = array (
    ' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ image_wate
rmark_obje ct )
);
$ image_wate rmark = array (
    ' WaterMarkT emplateId ' => $ watermark_
template_i d ,
    ' Type ' => ' Image ' ,
    ' InputFile ' => $ image_wate rmark_inpu t ,
    ' ReferPos ' => ' TopRight ' ,
    ' Width ' => 0 . 05 ,
    ' Dx ' => 0 ,
    ' Dy ' => 0
);
## Text Watermark
$ text_conf i g = array (
    ' Content ' => ' 5rWL6K + V5paH5a2X5 rC05Y2w ' ,
    ' FontName ' => ' SimSun ' ,
    ' FontSize ' => 16 ,
    ' FontColor ' => ' Red ' ,
```

```

        ' FontAlpha ' => 0 . 5 ,
        ' Top ' => 10 ,
        ' Left ' => 10
    );
$ text_water mark = array (
    ' WaterMarkT emplateId ' => $ watermark_
    template_i d ,
    ' Type ' => ' Text ',
    ' TextWaterM ark ' => $ text_conf_i g
);
## Video Watermark
$ video_wate rmark_inpu t = array (
    ' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ video_wate
rmark_obje ct )
);
$ video_wate rmark = array (
    ' WaterMarkT emplateId ' => $ watermark_
    template_i d ,
    ' Type ' => ' Image ',
    ' InputFile ' => $ video_wate rmark_inpu t ,
    ' ReferPos ' => ' BottomLeft ',
    ' Height ' => 240 ,
    ' Dx ' => 0 ,
    ' Dy ' => 0
);
# Output -> Watermarks
$ watermarks = array ($ image_wate rmark , $ text_water mark , $
video_wate rmark );
$ output [' WaterMarks '] = $ watermarks ;
# Outputs
$ outputs = array ($ output );
$ request -> setOutputs ( json_encod e ($ outputs ));
$ request -> setOutputB ucket ($ oss_bucket );
$ request -> setOutputL ocation ($ oss_locati on );
# PipelineId
$ request -> setPipel_i n eId ($ pipeline_i d );
# call api
try {
    $ response = $ client -> getAc_sResp onse ($ request );
    print ' RequestId is : ' . $ response ->{' RequestId ' } . "\ n
";
    if ($ response ->{' JobResultL ist '}->{' JobResult '}[ 0 ]->{'
Success '}) {
        print ' JobId is : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Job '}->{' JobId ' } . "\ n ";
    } else {
        print ' SubmitJobs Failed code : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Code ' } .
            ' message : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Message ' } . "\ n ";
    }
} catch ( ServerExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de () . ' Message : ' .
    $ e -> getMessage () . "\ n ";
} catch ( ClientExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de () . ' Message : ' .
    $ e -> getMessage () . "\ n ";
}

```

```
}
}
```

3.5.6 Screenshot

1. Create AcsClient instance.

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

2. Create request, and set parameters.

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

3. Set screenshot parameters.

· Input

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

· SnapshotConfig

- OutputFile

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

- Time

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

- Interval/Num

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

- Width/Height

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

4. Initiate API request and display returned value.

```
$ 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_000_01 . jpg ' . "\ n ";
```

```

print ' http ://'. $ oss_bucket .'.' $ oss_locati on .' aliyuncs
.com / output_000 02 . jpg ' . "\ n ";
print ' http ://'. $ oss_bucket .'.' $ oss_locati on .' aliyuncs
.com / output_000 03 . jpg ' . "\ n ";

```

Full code

```

<? php
include_on ce ' 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_i d = ' xxx ' ;
$ oss_locati on = ' oss - cn - hangzhou ' ;
$ oss_bucket = ' xxx ' ;
$ oss_input_ object = ' input . mp4 ' ;
$ oss_output _object = ' output_ { Count } . jpg ' ;
# DefaultAcs Client
$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id , # Region ID
    $ access_key _id , # AccessKey ID
    $ access_key _secret # AccessKey Secret
);
$ client = new DefaultAcs Client ($ clientProf ile );
# request
$ request = new Mts \ SubmitSnap shotJobReq uest ();
$ request -> setAcceptF ormat (' JSON ');
# Input
$ input = array (' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ oss_input_ object ));
$ request -> setInput ( json_encod e ($ input ));
# SnapshotCo nfig -> OutputFile
$ output = array (' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ oss_output _object ));
$ snapshot_c onfig = array (' OutputFile ' => $ output );
# SnapshotCo nfig -> Time
$ snapshot_c onfig [' Time '] = 2 ;
# SnapshotCo nfig -> Interval / Num
$ snapshot_c onfig [' Interval '] = 2 ;
$ snapshot_c onfig [' Num '] = 3 ;
# SnapshotCo nfig -> Width / Height
$ snapshot_c onfig [' Height '] = 360 ;
# SnapshotCo nfig
$ request -> setSnapsho tConfig ( json_encod e ($ snapshot_c onfig
));
# PipelineId
$ request -> setPipelin eId ($ pipeline_i d );
# call api
try {
    $ response = $ client -> getAcsResp onse ($ request );
    print ' RequestId is : ' . $ response ->{' RequestId ' } . "\ n
";
    print ' JobId is : ' . $ response ->{' SnapshotJo b ' }->{' Id
'} . "\ n ";
    print ' http ://'. $ oss_bucket .'.' $ oss_locati on .'
aliyuncs . com / output_000 01 . jpg ' . "\ n ";
    print ' http ://'. $ oss_bucket .'.' $ oss_locati on .'
aliyuncs . com / output_000 02 . jpg ' . "\ n ";

```

```

    print ' http ://'. $ oss_bucket .'.'. $ oss_locati on .'
    aliyuncs . com / output_000 03 . jpg ' . "\ n ";
} catch ( ServerExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
} catch ( ClientExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
}

```

3.5.7 Splicing and simple cutting

1. Create AcsClient instance.

```

$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id ,           # Region ID
    $ access_key _id ,          # AccessKey ID
    $ access_key _secret       # AccessKey Secret
);
$ client = new DefaultAcs Client ( $ clientProf ile );

```

2. Create request, and set parameters.

```

$ request = new Mts \ SubmitJobs Request ();
$ request -> setAcceptF ormat ( ' JSON ' );

```

3. Set transcoding parameters.

• Input

```

$ input = array ( ' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ( $ head_objec t ));
$ request -> setInput ( json_encod e ( $ input ));

```

• Output

```

$ output = array ( ' OutputObje ct ' => urlencode ( $
    oss_output _object ));

```

- Video

```

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

```

- MergeList

```

$ merge_vide o = array ( ' MergeURL ' => ' http ://'. $
    oss_bucket .'.'. $ oss_locati on .' aliyuncs . com /'.
    urlencode ( $ oss_input_ object ));
$ merge_tail = array ( ' MergeURL ' => ' http ://'. $
    oss_bucket .'.'. $ oss_locati on .' aliyuncs . com /'.
    urlencode ( $ tail_objec t ));

```

```
$ output [' MergeList ' ] = array ( $ merge_vide o , $
merge_tail );
```

4. Initiate API request and display returned value.

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

Full codes

```
include_on ce ' 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_i d = ' xxx ' ;
$ template_i d = ' S00000001 - 200030 ' ;
$ oss_locati on = ' oss - cn - hangzhou ' ;
$ oss_bucket = ' xxx ' ;
$ oss_input_ object = ' input . mp4 ' ;
$ oss_output _object = ' output . mp4 ' ;
$ head_objec t = ' head . mp4 ' ;
$ tail_objec t = ' tail . mp4 ' ;
// Create a DefaultAcS Client instance and initialize it
;
$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id , # Region ID
    $ access_key _id , # AccessKey ID
    $ access_key _secret # AccessKey Secret
);
$ client = new DefaultAcS Client ( $ clientProf ile );
# Create an API request and set parameters
$ request = new Mts \ SubmitJobs Request ();
$ request -> setAcceptF ormat ( ' JSON ' );
# Input
$ input = array ( ' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ( $ head_objec t ));
$ request -> setInput ( json_encod e ( $ input ));
# Output
$ output = array ( ' OutputObje ct ' => urlencode ( $ oss_output
_object ));
# Ouput -> Video
$ output [' Video ' ] = array ( ' Width ' => 1280 ,
    ' Height ' => 720 );
# Ouput -> TemplateId
$ output [' TemplateId ' ] = $ template_i d ;
```

```

# 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_objec t ));
$ output [ ' MergeList ' ] = array ( $ merge_video , $ merge_tail );
# Outputs
$ outputs = array ( $ output );
$ request -> setOutputs ( json_encod e ( $ outputs ));
$ request -> setOutputB ucket ( $ oss_bucket );
$ request -> setOutputL ocation ( $ oss_locati on );
# PipelineId
$ request -> setPipelin eId ( $ pipeline_i d );
# call api
try {
    $ response = $ client -> getAc sResp onse ( $ request );
    print ' RequestId is : ' . $ response ->{ ' RequestId ' } . "\ n
";
    if ( $ response ->{ ' JobResultL ist ' }->{ ' JobResult ' }[ 0 ]->{ '
Success ' }) {
        print ' JobId is : ' .
            $ response ->{ ' JobResultL ist ' }->{ ' JobResult ' }[ 0
]->{ ' Job ' }->{ ' JobId ' } . "\ n ";
    } else {
        print ' SubmitJobs Failed code : ' .
            $ response ->{ ' JobResultL ist ' }->{ ' JobResult ' }[ 0
]->{ ' Code ' } .
            ' message : ' .
            $ response ->{ ' JobResultL ist ' }->{ ' JobResult ' }[ 0
]->{ ' Message ' } . "\ n ";
    }
} catch ( ServerExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
} catch ( ClientExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
}
}

```

3.5.8 Splicing-opening and ending scenes

1. Create Ac sClient instance.

```

$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id ,           # Region ID
    $ access_key _id ,          # AccessKey ID
    $ access_key _secret        # AccessKey Secret
);
$ client = new DefaultAc s Client ( $ clientProf ile );

```

2. Create request, and set parameters.

```

$ request = new Mts \ SubmitJobs Request ();

```

```
$ request -> setAcceptFormat ( ' JSON ' );
```

3. Set transcoding parameters.

· Input

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

· Output

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

- Video

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

- OpeningList

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

- TailSlateList

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

4. Initiate API request and display returned value.

```
$ 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 ";
```



```
}

```

Full codes

```
<? 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_i d = ' xxx ' ;
$ template_i d = ' S00000001 - 200030 ' ;
$ oss_locati on = ' oss - cn - hangzhou ' ;
$ oss_bucket = ' xxx ' ;
$ oss_input_ object = ' input . mp4 ' ;
$ oss_output_ object = ' output . mp4 ' ;
$ head_objec t = ' head . mp4 ' ;
$ tail_objec t = ' tail . mp4 ' ;
# Create DefaultAcs Client instance and complete
initializa tion
$ clientProf ile = DefaultPro file :: getProfile (
    $ mps_region _id , # Region ID
    $ access_key _id , # AccessKey ID
    $ access_key _secret # AccessKey Secret
);
$ client = new DefaultAcs Client ($ clientProf ile );
# Create API request and set parameters
$ request = new Mts \ SubmitJobs Request ();
$ request -> setAcceptF ormat (' JSON ');
# Input
$ input = array (' Location ' => $ oss_locati on ,
    ' Bucket ' => $ oss_bucket ,
    ' Object ' => urlencode ($ oss_input_ object ));
$ request -> setInput ( json_encod e ($ input ));
# Output
$ output = array (' OutputObje ct ' => urlencode ($ oss_output_
_object ));
# Output -> TemplateId
$ output [' TemplateId '] = $ template_i d ;
# Output -> OpeningLis t
$ opening_vi deo = array (' OpenUrl ' => ' http ://'. $ oss_bucket
.'. '$ oss_locati on .' . aliyuncs . com /'. urlencode ($ head_objec
t ),
    ' Width ' => 640 ,
    ' Start ' => 2 );
$ output [' OpeningLis t '] = array ($ opening_vi deo );
# Output -> TailSlateL ist
$ tailslate_ video = array (' TailUrl ' => ' http ://'. $ oss_bucket
.'. '$ oss_locati on .' . aliyuncs . com /'. urlencode ($ tail_objec
t ),
    ' Width ' => 640 ,
    ' BlendDurat ion ' => 3 ,
    ' BgColor ' => ' Black ');
$ output [' TailSlateL ist '] = array ($ tailslate_ video );
# Outputs
$ outputs = array ($ output );
$ request -> setOutputs ( json_encod e ($ outputs ));
$ request -> setOutputB ucket ($ oss_bucket );
$ request -> setOutputL ocation ($ oss_locati on );
# PipelineId
$ request -> setPipelin eId ($ pipeline_i d );
# call api
```

```

try {
    $ response = $ client -> getAcqsResp onse ($ request );
    print ' RequestId is : ' . $ response ->{' RequestId ' } . "\ n
";
    if ($ response ->{' JobResultL ist '}->{' JobResult '}[ 0 ]->{'
Success '}) {
        print ' JobId is : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Job '}->{' JobId ' } . "\ n ";
    } else {
        print ' SubmitJobs Failed code : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Code ' } ;
        ; message : ' .
            $ response ->{' JobResultL ist '}->{' JobResult '}[ 0
]->{' Message ' } . "\ n ";
    }
} catch ( ServerExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
} catch ( ClientExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
}

```

3.5.9 Create HLS standard encryption workflow

Overview

The article is an example of calling API to create HLS standard encryption workflow.

For more information about creating workflow, see [AddMediaWorkflow](#). For more information about MPS SDK, see [Installation](#).

Code example

```

<? php
include_on ce ' aliyun - php - sdk - core / Config . php ' ;
use Mts \ Request \ V20140618 as Mts ;
date_defau lt_timezon e_set ( ' PRC ' );
class HLSEncrypt ionWorkflo wDemo {
    private $ client ;
    private $ region = '< region >';
    private $ accessKeyI d = '< accessKeyI d >';
    private $ accessKeyS ecret = '< accessKeyS ecret >';
    private $ pipelineId = "< PipelineId >";
    private $ templateId = " S00000001 - 100020 "; # transcodin g
template ID , m3u8 template , set as needed
    private $ ossLocatio n = "< OssLocatio n >";
    private $ inputBucke t = "< InputBucke t >";
    private $ inputPath = "< InputPath >"; # Example : " HLS -
Encryption "
    private $ outputBuck et = "< OutputBuck et >";
    private $ encryption Type = " hls - aes - 128 ";
    private $ hlsKeyUri = "< decryption key URI >"; #如 http
:// decrypt . testdomain . com
    private $ actStart = " Act - Start ";
    private $ actEncrypt ion = " Act - HLS - Encryption ";
    private $ actReport = " Act - Report ";
    function __construc t ( ) {

```

```

    $ profile = DefaultProfile::getProfile($this->region
, $this->accessKeyId, $this->accessKeySecret);
    $this->client = new DefaultAcsClient($profile);
}
function addMediaWorkflow() {
    $request = new Mts\AddMediaWorkflowRequest();
    $request->setName("HLS encryption workflow 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 () {
    $ transcodingParameters = array (
        " OutputBucket " => $ this -> outputBucket ,
        " OutputLocation " => $ this -> ossLocation ,
        " Outputs " => $ this -> buildOutputsConfig ()
    );
    return $ transcodingParameters ;
}
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 -> encryption Type ,
        " KeyUri " => $ this -> hlsKeyUri
    );
    return $ encryption ;
}
function buildReportActivity () {
    $ reportActivity = array (
        " Name " => $ this -> actReport ,
        " Parameters " => ( object )[],
        " Type " => " Report "
    );
    return $ reportActivity ;
}
function buildDependencies () {
    $ subActivityOfStart = array (
        $ this -> actEncryption
    );
    $ subActivityOfTranscode = array (
        $ this -> actReport
    );
    $ dependencies = array (
        $ this -> actStart => $ subActivityOfStart ,
        $ this -> actEncryption => $ subActivityOfTranscode
    ,
        $ this -> actReport => []
    );
    return $ dependencies ;
}
}
$demo = new HlsEncryptionWorkflowDemo ();
$demo -> addMediaWorkflow ();

```

? >

3.5.10 MPS queue management

The system automatically creates an MPS queue when you open the MPS service.

You can also use interfaces to manage MPS queue (pipeline). For example,

SearchPipeline, QueryPipelineList, UpdatePipeline.

Search MPS queue

You can use the SearchPipeline interface to search the MPS queue information.

```
$ region = '< region >';
  $ accessKeyId = '< accessKeyId >';
  $ accessKeySecret = '< accessKeySecret >';
  $ profile = DefaultProfile::getProfile($ region, $
accessKeyId, $ accessKeySecret);
  $ client = new DefaultAcsClient($ profile);
  $ request = new Mts\SearchPipelineRequest();
  // If an error occurs, it can throw ClientException
  // or ServerException.
  $ response = $ client->getAcsResponse($ request);
  $ pipelines = $ response->PipelineList->Pipeline;
  foreach ($ pipelines as $ pipeline) {
    echo 'pipeline id:' . $ pipeline->Id . ', name:' .
$ pipeline->Name . ', state:' . $ pipeline->State . "\n";
  }
```

Query MPS queue

If you have the pipelineId, you can use pipelineId to call QueryPipelineList interface to query MPS queue information.

```
$ region = '< region >';
  $ accessKeyId = '< accessKeyId >';
  $ accessKeySecret = '< accessKeySecret >';
  // The known pipeline ID, separated by comma (,)
  $ pipelineIds = '< pipelineIds >';
  $ profile = DefaultProfile::getProfile($ region, $
accessKeyId, $ accessKeySecret);
  $ client = new DefaultAcsClient($ profile);
  $ request = new Mts\QueryPipelineListRequest();
  $ request->setPipelineIds($ pipelineIds);
  // If an error occurs, it can throw ClientException
  // or ServerException.
  $ response = $ client->getAcsResponse($ request);
  $ pipelines = $ response->PipelineList->Pipeline;
  foreach ($ pipelines as $ pipeline) {
    echo 'pipeline id:' . $ pipeline->Id . ', name:' .
$ pipeline->Name . ', state:' . $ pipeline->State . "\n";
  }
```

```
}
}
```

Update MPS queue

Use the `UpdatePipeline` interface to update MPS queue information, including MPS queue name and status. The status includes `Active` and `Paused`.

```
$ region = '< region >';
$ accessKeyId = '< accessKeyId >';
$ accessKeySecret = '< accessKeySecret >';
$ profile = DefaultProfile::getProfile($ region, $
accessKeyId, $ accessKeySecret);
$ client = new DefaultAcsClient($ profile);
$ request = new Mts\SearchPipelineRequest();
// If an error occurs, it can throw ClientException
or ServerException.
$ 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 Query media - Use OSS file address

If you do not know the media ID (a live video converted to an on-demand video using the media workflow), you can use the media input URL to query the media information over `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 must be RFC3986 - encoded .
        $ ossObject = $ this->encodeByRFC3986('test / The
Legend of the Swordsman.mp4');
```

```

    $ request -> setFileURL s ($ ossDomain .$ ossObject );
    $ response = $ this -> client -> getAcResp onse ($ request
);
    echo json_encod e ($ response );
}
function encodeByRF C3986 ($ arg_1 )
{
    $ encodeOssO bject = "";
    $ arraylist = explode ("/", $ arg_1 );
    for ($ i = 0 ; $ i < count ($ arraylist ); $ i ++ )
    {
        $ tmp = rawurlenco de ($ arraylist [$ i ]);
        $ encodeOssO bject = $ encodeOssO bject .$ tmp ;
        if ($ i != count ($ arraylist ) - 1 ) {
            $ encodeOssO bject = $ encodeOssO bject ."/";
        }
    }
    return $ encodeOssO bject ;
}
}
$ demo = new QueryMedia ListByUrlD emo ();
$ demo -> queryMedia ListByUrl ();
? >

```

3.5.12 Add media

Add a video file to Media files:

```

<? php
include_on ce ' 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 ";
$ mtsEndpoin t = " mts ." + REGION + ". aliyuncs . com ";
# Step 2 . set accesskey & keySecret
$ accessKeyI d = "";
$ accessKeyS ecret = "";
# Create DefaultAcs Client instance and perform
initializa tion
$ clientProf ile = DefaultPro file :: getProfile (
    $ REGION , # Your Region ID
    $ accessKeyI d , # Your AccessKey ID
    $ accessKeyS ecret # Your AccessKey Secret
);
$ client = new DefaultAcs Client ($ clientProf ile );
$ request = new Mts \ AddMediaRe quest ();
$ request -> setAcceptF ormat (' JSON ');
$ request -> setFileURL (" http :// mtb - sz - in . oss - cn -
shenzhen . aliyuncs . com / media / r180 - ABC . mp4 ");
$ request -> setMediaWo rkflowId (" 829bed0300 994057a49e
4f16de957e 34 ");
# Initiate request and handle returned result
try {
    $ response = $ client -> getAcResp onse ($ request );
    print ' RequestId is : ' . $ response ->{' RequestId ' } . "\ n
";
    print " Response :". json_encod e ($ response );
} catch ( ServerExce ption $ e ) {
    print ' Error : ' . $ e -> getErrorCo de () . ' Message : ' .
$ e -> getMessage () . "\ n ";

```

```

} catch ( ClientException $ e ) {
    print ' Error : ' . $ e -> getErrorCo de ( ) . ' Message : ' .
    $ e -> getMessage ( ) . "\ n ";
}
? >

```

3.5.13 Add media workflow

The user can assemble activities provided by MPS, such as transcode activity and screenshot activity into a topology. The topology is as follows:

```

<? php
include_on ce ' 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 ";
$ mtsEndpoin t = " mts ." + REGION + ". aliyuncs . com ";
# Step 2 . set accesskey & keySecret
$ accessKeyI d = "";
$ accessKeyS ecret = "";
# Step 3 . set mps transcodin g queue id
$ PIPELINE_I D = " 38bba54d52 4448be92d2 77caaa8da1 18 ";
# Create DefaultAcs Client instance and perform initializa
tion
$ clientProf ile = DefaultPro file :: getProfile (
    $ REGION , # Your Region ID
    $ accessKeyI d , # Your AccessKey ID
    $ accessKeyS ecret # Your AccessKey Secret
);
$ client = new DefaultAcs Client ( $ clientProf ile );
$ request = new Mts \ AddMediaWo rkflowRequ est ( );
$ request -> setAcceptF ormat ( ' JSON ' );
$ request -> setName ( " Sequential - workflow " );
$ startActiv ity = array (
    " Type " => " Start ",
    " Parameters " => array (
        " InputFile " => array (
            " Bucket " => " mtb - sz - in ",
            " Location " => $ OSS_REGION ,
            " ObjectPref ix " => " media / "
        ),
        " PipelineId " => $ PIPELINE_I D
    )
);
$ transcodeA ctivity = array (
    " Type " => " Transcode ",
    " Parameters " => array (
        " Outputs " => array (
            array (
                " OutputObje ct " => urlencode ( " transcode / {
ObjectPref ix } / { FileName } . { ExtName } " ),
                " TemplateId " => " S00000001 - 000070 "
            )
        ),
        " OutputLoca tion " => $ OSS_REGION ,
        " OutputBuck et " => " mtb - sz - out "
    )
);
$ reportActi vity = array (
    " Type " => " Report ",

```



```

        " Parameters "=> array (
            " PublishTyp e "=>" Auto "
        )
    );
    $ topology = array (
        " Activities "=> array (
            " startNode "=>$ startActiv ity ,
            " transcodin gNode "=>$ transcodeA ctivity ,
            " reportNode "=>$ reportActi vity
        ),
        " Dependenci es "=> array (
            " startNode "=> array (" transcodin gNode "),
            " transcodin gNode "=> array (" reportNode "),
            " reportNode "=> array ()
        )
    );
    $ request -> setTopolog y ( json_encod e ($ topology ));
    # Initiate request and handle returned result
    try {
        $ response = $ client -> getAcsResp onse ($ request );
        print ' RequestId is : ' . $ response ->{' RequestId '} . "\ n
";
        print " Response :". json_encod e ($ response );
    } catch ( ServerExce ption $ e ) {
        print ' Error : ' . $ e -> getErrorCo de () . ' Message : ' .
        $ e -> getMessage () . "\ n ";
    } catch ( ClientExce ption $ e ) {
        print ' Error : ' . $ e -> getErrorCo de () . ' Message : ' .
        $ e -> getMessage () . "\ n ";
    }
? >

```

4 Demo project

Currently, Alibaba Cloud provides the MPS SDKs for Java, Python, and PHP. For details about the installation and usage instructions, see the following URLs:

- [Multi-region support](#)
- [Java](#)
- [PHP](#)
- [Python](#)

[Demo project](#)