



智能媒体管理 SDK参考

文档版本: 20220330



法律声明

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

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

通用约定

格式	说明	样例
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。	♪ 危险 重置操作将丢失用户配置数据。
▲ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔〕) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等 <i>,</i> 不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面,单击 确定 。
Courier字体	命令或代码。	执行 cd /d C:/window 命令 <i>,</i> 进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]
{} 或者 {a b}	表示必选项,至多选择一个。	switch {act ive st and}

目录

1.Java SDK	05
2.Node.js SDK	06
3.PHP SDK	07
4.Python SDK	<mark>0</mark> 8
5.Go SDK	12
6NET SDK	16

1.Java SDK

本文介绍如何安装和快速使用智能媒体管理Java SDK。

前提条件

- 已下载并安装J2SE 6.0及以上版本的Java开发环境。请通过Java官方网站下载相应版本。
- 已创建并获取Access Key。具体操作,请参见获取AccessKey。
- 已开通OSS服务、创建存储空间并上传文件到存储空间。具体操作,请参见OSS控制台使用流程。
- 已开通智能媒体管理服务。具体操作,请参见开通产品。

↓ 注意 目前在华东1(杭州)和华东2(上海)地域中,您也可以通过调用OpenImmService接口 开通智能媒体管理服务。

• 已创建所需类型的项目。具体操作,请参见创建项目。

快速使用

使用智能媒体管理Java SDK:

1. 安装智能媒体管理Java SDK。

SDK安装和使用帮助文档: https://github.com/aliyun/aliyun-openapi-java-sdk/blob/master/README-CN.md

2. API调试和SDK示例。

阿里云API调试工具提供 OpenAPI 的文档、API调试、SDK示例,帮助您快速入门 API 开发。

API调试工具: https://next.api.aliyun.com/api/imm/2017-09-06/

2.Node.js SDK

本文介绍如何安装和快速使用智能媒体管理Node.js SDK。

前提条件

- 已下载并安装Node.js 6.x以上版本的开发环境。
- 已创建并获取Access Key。具体操作,请参见获取AccessKey。
- 已开通OSS服务、创建存储空间并上传文件到存储空间。具体操作,请参见OSS控制台使用流程。
- 已开通智能媒体管理服务。具体操作,请参见开通产品。

↓ 注意 目前在华东1(杭州)和华东2(上海)地域中,您也可以通过调用OpenImmService接口 开通智能媒体管理服务。

• 已创建所需类型的项目。具体操作,请参见创建项目。

快速使用

使用智能媒体管理Node.js SDK:

- 1. SDK安装和使用帮助文档: https://github.com/aliyun/openapi-core-nodejs-sdk
- 2. API调试和SDK示例。

阿里云API调试工具提供 OpenAPI 的文档、API调试、SDK示例,帮助您快速入门 API 开发。 API调试工具: https://next.api.aliyun.com/api/imm/2017-09-06/

3.PHP SDK

本文介绍如何安装和快速使用智能媒体管理PHP SDK。

前提条件

- 已下载并安装PHP 5.5及以上版本的开发环境。请通过PHP官方网站下载相应版本。
- 已创建并获取Access Key。具体操作,请参见获取AccessKey。
- 已开通OSS服务、创建存储空间并上传文件到存储空间。具体操作,请参见OSS控制台使用流程。
- 已开通智能媒体管理服务。具体操作,请参见开通产品。

↓ 注意 目前在华东1(杭州)和华东2(上海)地域中,您也可以通过调用OpenImmService接口 开通智能媒体管理服务。

• 已创建所需类型的项目。具体操作,请参见创建项目。

快速使用

使用智能媒体管理PHP SDK:

1. 安装智能媒体管理PHP SDK。

SDK安装和使用帮助文档: https://github.com/aliyun/openapi-sdk-php/blob/master/README-zh-CN.md

2. API调试和SDK示例。

阿里云API调试工具提供 OpenAPI 的文档、API调试、SDK示例,帮助您快速入门 API 开发。

AP调试工具: https://next.api.aliyun.com/api/imm/2017-09-06/

4.Python SDK

本文介绍如何安装和快速使用智能媒体管理Python SDK。

前提条件

- 已下载并安装Python 2或Python 3的开发环境。
- 已创建并获取Access Key。具体操作,请参见获取AccessKey。
- 已开通OSS服务、创建存储空间并上传文件到存储空间。具体操作,请参见OSS控制台使用流程。
- 已开通智能媒体管理服务。具体操作,请参见开通产品。

↓ 注意 目前在华东1(杭州)和华东2(上海)地域中,您也可以通过调用OpenImmService接口 开通智能媒体管理服务。

• 已创建所需类型的项目。具体操作,请参见创建项目。

快速使用

使用智能媒体管理Python SDK的操作步骤如下:

1. 安装智能媒体管理Python SDK。

Python SDK的有如下两种安装方式。

○ 方式一: 通过pip安装(推荐)

执行如下命令通过pip安装智能媒体管理Python SDK。

pip install aliyun-python-sdk-core # 安装阿里云SDK核心库。
pip install aliyun-python-sdk-imm # 安装阿里云智能媒体管理Python SDK。

○ 方式二: 通过GithHub源码安装

如果未安装Git,请安装Git后,执行如下命令通过GitHub源码安装智能媒体管理Python SDK。

Python SDK在GitHub中的路径请参见Python SDK。

```
git clone https://github.com/aliyun/aliyun-openapi-python-sdk.git
# 安装阿里云SDK核心库。
cd aliyun-python-sdk-core
python setup.py install
# 安装阿里云智能媒体管理Python SDK。
cd aliyun-python-sdk-imm
python setup.py install
```

2. 开始一个Python项目。

调用Python SDK的主要步骤如下:

i. 创建Client实例。

在创建Client实例时,您需要获取并配置Region ID、AccessKey ID和AccessKey Secret。

- ii. 创建API请求并设置参数。
- iii. 发起请求并处理应答或异常。

如下示例以在华东1(杭州)地域中创建文档转换任务为例说明。

i. 创建文档转换任务。

```
#!/usr/bin/env python
#coding=utf-8
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkimm.request.v20170906.CreateOfficeConversionTaskRequest import Create
OfficeConversionTaskRequest
client = AcsClient('your access key id', 'your access key secret', 'cn-hangzhou')
createReq = CreateOfficeConversionTaskRequest()
srcUri = "oss://your-bucket-name/your-path/your-doc.doc"
tgtUri = "oss://your-bucket-name/your-path/your-doc-output"
tgtType = "vector"
createReq.set Project("immdoctest")
createReq.set SrcUri(srcUri)
createReq.set_TgtUri(tgtUri)
createReq.set TgtType(tgtType)
response = client.do_action_with_exception(createReq)
print (response)
```

任务创建成功的返回结果如下:

```
{
   "Status": "Running",
   "CreateTime": "2018-07-21T15:37:28.539Z",
   "RequestId": "01E3E9D5-48B2-4326-BC75-9AAD22953077",
   "Percent": 0,
   "TaskId": "ee52a217-234a-4922-8bd5-fcce5584c0e7",
   "TgtLoc": "oss://your-bucket-name/your-path/your-doc-output"
}
```

ii. 轮询获取文档转换任务状态,轮询时间间隔为1s(不能低于0.2s),超时时间为10s。

```
res = json.loads(response)
taskId = res["TaskId"]
getReq = GetOfficeConversionTaskRequest.GetOfficeConversionTaskRequest()
getReq.set Project("immdoctest")
getReq.set TaskId(taskId)
period = 1
timeout = 30
start = time.time()
while True:
   time.sleep(period)
   response = client.do action with exception(getReq)
   print (response)
   status = json.loads(response)["Status"]
   if status == "Finished": #任务完成。
      print ("Task finished.")
       break
   if status == "Failed":
                             #任务失败。
      print ("Task failed.")
       break
   if time.time() - start > timeout: #任务超时。
       print ("Task timeout.")
       break
```

当文档转换任务的执行状态为Running时的返回结果如下:

```
{
 "NotifyTopicName": "",
 "FailDetail": {
   "Code": ""
 },
 "TaskId": "ee52a217-234a-4922-8bd5-fcce5584c0e7",
 "NotifyEndpoint": "",
 "PageCount": 0,
 "Status": "Running",
 "TgtType": "vector",
 "FinishTime": "",
 "CreateTime": "2018-07-21T15:37:28.539Z",
 "RequestId": "4153D465-45D3-4457-A273-DED40505CAD5",
 "SrcUri": "oss://your-bucket-name/your-path/your-doc.doc",
 "Percent": 0,
 "TgtUri": "oss://your-bucket-name/your-path/your-doc-output"
```

当文档转换任务的执行状态为Finished时的返回结果如下:

```
{
 "NotifyTopicName": "",
  "FailDetail": {
   "Code": "NoError"
  },
  "TaskId": "ee52a217-234a-4922-8bd5-fcce5584c0e7",
  "NotifyEndpoint": "",
  "PageCount": 32,
  "Status": "Finished",
  "TgtType": "vector",
  "FinishTime": "2018-07-21T15:37:32.419Z",
  "CreateTime": "2018-07-21T15:37:28.539Z",
  "RequestId": "CACA2EC6-98FC-457A-810F-4035DD32A43B",
  "SrcUri": "oss://your-bucket-name/your-path/your-doc.doc",
 "Percent": 100,
 "TgtUri": "oss://your-bucket-name/your-path/your-doc-output"
}
```

当文档转换任务的执行状态为Failed时的返回结果如下:

```
{
 "NotifyTopicName": "",
 "FailDetail": {
   "Code": "FailedToDownloadSrcFile"
 },
 "TaskId": "1c8d3a19-9f28-433f-be13-8e42b8851ef3",
 "NotifyEndpoint": "",
 "PageCount": 0,
 "Status": "Failed",
  "TgtType": "vector",
  "FinishTime": "",
 "CreateTime": "2018-07-21T15:47:53.865Z",
 "RequestId": "453B7362-D404-47A3-88A7-BE14F494812C",
 "SrcUri": "oss://your-bucket-name/your-path/your-doc.doc",
 "Percent": 0,
 "TgtUri": "oss://your-bucket-name/your-path/your-doc-output"
}
```

5.Go SDK

本文介绍如何安装和快速使用智能媒体管理Go SDK。

前提条件

- 已下载并安装Go的开发环境。
- 已创建并获取Access Key。具体操作,请参见获取AccessKey。
- 已开通OSS服务、创建存储空间并上传文件到存储空间。具体操作,请参见OSS控制台使用流程。
- 已开通智能媒体管理服务。具体操作,请参见开通产品。

↓ 注意 目前在华东1(杭州)和华东2(上海)地域中,您也可以通过调用OpenImmService接口 开通智能媒体管理服务。

• 已创建所需类型的项目。具体操作,请参见创建项目。

快速使用

使用智能媒体管理Go SDK的操作步骤如下:

1. 安装智能媒体管理Go SDK。

执行如下命令安装智能媒体管理Go SDK。

Go SDK在Git Hub中的路径请参见Go SDK。

go get github.com/aliyun/alibaba-cloud-sdk-go

安装过程中出现如下错误时,请忽略该错误。

can't load package: package github.com/aliyun/alibaba-cloud-sdk-go: no Go files in /Use
rs/zouyongbo/Projects/go-deps/src/github.com/aliyun/alibaba-cloud-sdk-go

2. 开始一个Go项目。

使用Go SDK的主要步骤如下:

i. 创建Client实例。

在创建Client实例时,您需要获取并配置Region ID、AccessKey ID和AccessKey Secret。

- ii. 创建API请求并设置参数。
- iii. 发起请求并处理应答或异常。

如下示例以在华东1(杭州)地域中创建文档转换任务为例说明。

```
package main
import (
    "fmt"
    "time"
    "github.com/aliyun/alibaba-cloud-sdk-go/services/imm"
)
func main() {
    //初始化IMM客户端。
    immClient, err := imm.NewClientWithAccessKey(
        "cn-hangzhou", //项目所在地域ID。
    "www.newcliential" //payEphacesyKey ID
```

```
"your_access_key_1d",
                             //KAMH/ PJACCESSKEY ID.
    "your_access_key_secret") //RAM用户的AccessKey Secret。
if err != nil {
   // Handle exceptions
   panic(err)
}
//项目名称
projectName := "imm-go-sdk-doc-demo"
//创建文档转换任务。
req2 := imm.CreateCreateOfficeConversionTaskRequest()
req2.Project = projectName
req2.SrcUri = "oss://your-bucket-name/your-path/your-doc.doc"
req2.TgtType = "vector"
req2.TgtUri = "oss://your-bucket-name/your-path/your-doc-output"
response2, err := immClient.CreateOfficeConversionTask(req2)
if err != nil {
   //Handle exceptions
   panic(err)
}
fmt.Println(response2)
//等待任务完成。
maxGetCount := 30
getInternval := time.Second
getCount := 0
req3 := imm.CreateGetOfficeConversionTaskRequest()
req3.Project = projectName
req3.TaskId = response2.TaskId
for {
    time.Sleep(getInternval)
   response3, err := immClient.GetOfficeConversionTask(req3)
    if err != nil {
       //Handle exceptions
       panic(err)
    }
    fmt.Println(response3)
    if response3.Status != "Running" {
       break
    }
    getCount++
    if getCount >= maxGetCount {
       fmt.Println("OfficeConversion Timeout for 30 seconds")
       break
    }
}
```

当文档转换任务的执行状态为Running时,返回结果如下:

}

```
{
 "NotifyTopicName": "",
 "FailDetail": {
   "Code": ""
 },
 "TaskId": "ee52a217-234a-4922-8bd5-fcce5584c0e7",
 "NotifyEndpoint": "",
 "PageCount": 0,
 "Status": "Running",
 "TgtType": "vector",
 "FinishTime": "",
 "CreateTime": "2018-07-21T15:37:28.539Z",
 "RequestId": "4153D465-45D3-4457-A273-DED40505CAD5",
 "SrcUri": "oss://your-bucket-name/your-path/your-doc.doc",
 "Percent": 0,
 "TgtUri": "oss://your-bucket-name/your-path/your-doc-output"
}
```

当文档转换任务的执行状态为Finished时,返回结果如下:

```
{
 "NotifyTopicName": "",
  "FailDetail": {
   "Code": "NoError"
 },
 "TaskId": "ee52a217-234a-4922-8bd5-fcce5584c0e7",
  "NotifyEndpoint": "",
 "PageCount": 32,
 "Status": "Finished",
 "TgtType": "vector",
  "FinishTime": "2018-07-21T15:37:32.419Z",
 "CreateTime": "2018-07-21T15:37:28.539Z",
 "RequestId": "CACA2EC6-98FC-457A-810F-4035DD32A43B",
 "SrcUri": "oss://your-bucket-name/your-path/your-doc.doc",
 "Percent": 100,
 "TgtUri": "oss://your-bucket-name/your-path/your-doc-output"
}
```

当文档转换任务的执行状态为Failed时,返回结果如下:

```
{
 "NotifyTopicName": "",
 "FailDetail": {
   "Code": "FailedToDownloadSrcFile"
 },
 "TaskId": "1c8d3a19-9f28-433f-be13-8e42b8851ef3",
 "NotifyEndpoint": "",
 "PageCount": 0,
 "Status": "Failed",
 "TgtType": "vector",
 "FinishTime": "",
 "CreateTime": "2018-07-21T15:47:53.865Z",
 "RequestId": "453B7362-D404-47A3-88A7-BE14F494812C",
 "SrcUri": "oss://your-bucket-name/your-path/your-doc.doc",
 "Percent": 0,
 "TgtUri": "oss://your-bucket-name/your-path/your-doc-output"
}
```

6..NET SDK

本文介绍如何安装和快速使用智能媒体管理.NET SDK。

前提条件

- 已下载并安装.NET Framework4.0及以上版本的开发环境。请通过.NET 官方网站下载相应版本。
- 已创建并获取Access Key。具体操作,请参见获取AccessKey。
- 已开通OSS服务、创建存储空间并上传文件到存储空间。具体操作,请参见OSS控制台使用流程。
- 已开通智能媒体管理服务。具体操作,请参见开通产品。

↓ 注意 目前在华东1(杭州)和华东2(上海)地域中,您也可以通过调用OpenImmService接口 开通智能媒体管理服务。

• 已创建所需类型的项目。具体操作,请参见创建项目。

快速使用

使用智能媒体管理.NET SDK的操作步骤如下:

1. 安装智能媒体管理.NET SDK。

如下以使用NuGet依赖安装智能媒体管理.NET SDK为例说明。其他安装智能媒体管理.NET SDK的方式, 请参见安装阿里云.NET SDK。

i. 创建一个文件夹demo。

mkdir demo

ii. 进入demo文件夹, 生成一个console工程。

```
cd demo
dotnet new console
```

生成的目录结构如下:

```
demo
|--demo.csproj
|--Program.cs
|-- obj/
```

iii. 将如下依赖加入工程下demo.csproj文件的 <Project> 元素中。

↓ 注意 请关注产品提供的最新版本.NET SDK, 查看相关功能并选择使用。

```
<ItemGroup>
```

```
<PackageReference Include="aliyun-net-sdk-core" Version="1.1.12"/>
<PackageReference Include="aliyun-net-sdk-imm" Version="1.0.0"/>
</ItemGroup>
```

2. 开始一个C#项目。

编辑 Program.cs 。如下示例以在华东1(杭州)地域中列出已创建的所有项目信息为例说明。

```
using System;
using Aliyun.Acs.Core;
using Aliyun.Acs.Core.Profile;
using Aliyun.Acs.Core.Exceptions;
using Aliyun.Acs.imm.Model.V20170906;
using System.Collections.Generic;
namespace net test
{
   class Program
    {
       static void Main(string[] args)
       {
           //构建一个Client,用于发起请求.
           IClientProfile profile = DefaultProfile.GetProfile(
                                    //项目所在地域ID。
               "cn-hangzhou",
               "your_access_key_id", //RAM用户的AccessKey ID。
               "your_access_key_secret"); //RAM用户的AccessKey Secret。
           DefaultAcsClient client = new DefaultAcsClient(profile);
           try
           {
               //构造请求。
               ListProjectsRequest request = new ListProjectsRequest();
               //发起请求并得到所有已创建的项目信息。
               ListProjectsResponse response = client.GetAcsResponse(request);
               List<Aliyun.Acs.imm.Model.V20170906.ListProjectsResponse.ListProjects P
rojectsItem> projects = response.Projects;
               foreach(Aliyun.Acs.imm.Model.V20170906.ListProjectsResponse.ListProject
s_ProjectsItem project in projects) {
                   Console.WriteLine(project.Project);
               }
           }
           catch (ServerException ex)
           {
               System.Console.WriteLine(ex.ToString());
           }
           catch (ClientException ex)
           {
               System.Console.WriteLine(ex.ToString());
           }
      }
   }
}
```