

Java上传SDK说明文档

Polyv Java 上传 SDK 为您提供服务端上传媒体文件到[保利威云点播平台](#)的开发工具包。

功能

- 快捷上传多种格式的媒体文件。
- 支持上传时的各种设置，如文件标题、描述、标签、上传目录、是否开启课件优化处理等。
- 采用分片并发上传的方式，支持断点续传。

使用方法

前提条件

1. 使用本 SDK 前，要先开通[保利威云点播服务](#)。如果您还不了解该服务，请登录产品主页查看，详见：[云点播](#)。
2. 获取userId、secretKey 等相关信息用于用户身份校验，您可以在「云点播管理后台 -> 设置 -> API 接口」页面中找到相关信息，[点击这里登录后台](#)。

环境要求

Java版本1.8.0及以上

集成 SDK

1. 下载Java上传SDK示例代码polyv-upload-java-sdk-1.1.1.zip开发包(包含示例代码和所需jar包)；
2. 将解压后的jar文件拷贝至您的项目中；
3. 使用IDE开发时引用jar包的方法，以Eclipse和IntelliJ IDEA为例说明如下：

3.1 在Eclipse中选择您的工程，右击 -> Properties -> Java Build Path -> Add JARs;

3.2 在IntelliJ IDEA中打开您的工程，File -> Project Structure -> Modules -> 右侧Dependencies -> + -> JARs or directories

API参考

1. 入口类 PolyvUploadClient.java

1.1 构造方法

PolyvUploadClient(String userId, String secretKey, int partitionSize, String checkpoint, int threadNum)

参数解释：

参数	说明
userId	保利威账号ID
secretKey	保利威账号的secretKey
partitionSize	可指定分片上传时每个分片的大小，单位：字节，默认为1MB，取值范围为100KB~5GB
checkpoint	使用断点续传时，写入续传位置的本地磁盘路径，需保证文件路径已经创建好
threadNum	可指定分片上传时的并发线程数，默认为1。注：该配置会占用服务器CPU资源，需根据服务器情况指定。

1.2 上传方法

uploadVideoParts(VideoInfo videoInfo)

该方法返回视频ID。

2 视频实体类 VideoInfo.java

视频信息字段：

名称	类型	必填	描述
fileLocation	String	Y	视频文件在本地的绝对路径，必须包含文件后缀
title	String	N	视频标题，为空时会取视频文件名作为标题（不含后缀）
fileSize	Long	N	文件大小
describ	String	N	视频简介
tag	String	N	视频标签，多个用,分隔
catald	Long	N	视频分类ID，默认值：1，表示上传到默认分类
luping	Int	N	是否进行 课件优化 处理，默认值：0
keepsouce	Int	N	是否保持源文件播放，默认：0
videoPoolId	String	N	视频ID，当上传中断续传时，需要指定vid来恢复上传
state	String	N	自定义拓展信息，如果提交了该字段，会在上传完成的 事件回调 中透传返回。

示例代码

```

import net.polyv.bean.vo.VideoInfo;
import net.polyv.callback.UploadCallBack;
import net.polyv.entry.PolyvUploadClient;
import net.polyv.enumeration.UploadErrorMsg;

public class PolyvUploadVideoDemo {
    private static final String userId = ""; //保利威账号的userId(必填)
    private static final String secretKey = ""; //账号的secretKey(必填)

    public static void main(String[] args) {
        int partitionSize = 1*1024 * 1024; //可指定分片上传时每个分片的大小，默认为1M
        int threadNum = 1; //可指定分片上传时的并发线程数，默认为1，(注：该配置会占用服务器CPU资源，需根据服务器情况指定)
        String checkpoint = "/Users/TestUser/Desktop"; //断点续传的上传位置写入路径
        PolyvUploadClient client = new PolyvUploadClient(userId, secretKey,
partitionSize, checkpoint, threadNum); //初始化PolyvUploadClient实例

        VideoInfo videoInfo = new VideoInfo();
        videoInfo.setFileLocation("/Users/TestUser/Desktop/testVideo.mp4"); //
视频文件在服务器上的绝对路径，必须包含拓展名（必填）
        videoInfo.setTitle("测试视频"); // 视频标题（必填）
        videoInfo.setCataId(1L); // 上传的分类目录，默认值：1，表示上传到默认分类（可
选）
        videoInfo.setDescrib("视频描述"); // 视频描述（可选）
        videoInfo.setTag("视频标签"); // 视频标签（可选）
        videoInfo.setLuping(0); //是否进行课件优化处理，默认值：0（可选）
        videoInfo.setKeepSource(0); //是否保持源文件播放，默认：0（可选）
        videoInfo.setState("自定义拓展信息"); //如果提交了该字段，会在上传完成的事件回
调中透传返回

        System.out.println("vid=" + client.uploadVideoParts(videoInfo, new
UploadCallBack() {
            @Override
            public void start(String s) {
                System.out.println("start=" + s);
            }
            @Override
            public void process(String s, long l, long l1) {
                System.out.println("process=" + s + ",uploaded=" + l + ",
total=" + l1);
            }
            @Override
            public void complete(String s) {
                System.out.println("complete=" + s);
            }
            @Override
            public void success(String s) {
                System.out.println("success=" + s);
            }
        }

```

```

        @Override
        public void error(String s, UploadErrorMsg uploadErrorMsg) {
            System.out.println("error=" + s + ", message=" +
uploadErrorMsg.getMessage());
        }
    }, true));
    // 当上传中断, 重新恢复的时候, 可以指定vid来恢复上传
videoInfo.setVideoPoolId("xxxxxxxxxxx");
    //      videoInfo = new VideoInfo();
    //
videoInfo.setFileLocation("C:\\Users\\Lenovo\\Desktop\\video\\test.mp4");
    //      videoInfo.setVideoPoolId("xxxxxxxxxxx");
    //      System.out.println("vid=" + client.uploadVideoParts(videoInfo),
new UploadCallBack() {
    //          @Override
    //          public void start(String s) {
    //              System.out.println("start=" + s);
    //          }
    //          @Override
    //          public void process(String s, long l, long ll) {
    //              System.out.println("process=" + s + ",uploaded=" + l +
", total=" + ll);
    //          }
    //          @Override
    //          public void complete(String s) {
    //              System.out.println("complete=" + s);
    //          }
    //          @Override
    //          public void success(String s) {
    //              System.out.println("success=" + s);
    //          }
    //          @Override
    //          public void error(String s, UploadErrorMsg uploadErrorMsg)
{
    //              System.out.println("error=" + s + ", message=" +
uploadErrorMsg.getMessage());
    //          }
    //      }, true);
    //  }
}
}

```

Change log

日期	版本	更新内容
2020-03-29	v1.1.1	支持自定义拓展字段，可在上传完成回调中透传返回
2019-08-05	v1.0.0	支持断点续传的新版Java上传SDK 发布