

裁剪录制文件

接口描述

- 1、裁剪直播录制视频文件，裁剪文件过程为异步处理过程
- 2、裁剪视频的最小粒度仅支持到单位为秒(s)的裁剪
- 3、接口支持https协议

接口URL

http://api.polyv.net/live/v3/channel/record/clip

在线API调用

请求方式

POST

接口约束

1、接口同时支持HTTP 、HTTPS ， 建议使用HTTPS 确保接口安全， 接口调用有频率限制， [详细请查看](#)

请求参数描述

参数名	必选	类型	说明
appId	true	String	账号appId【详见 获取密钥 】
timestamp	true	Long	当前13位毫秒级时间戳，3分钟内有效

参数名	必选	类型	说明
sign	true	String	签名，为32位大写的MD5值, 生成签名的appSecret密钥作为通信数据安全的关键信息，严禁保存在客户端直接使用，所有API都必须通过客户自己服务器中转调用POLYV服务器获取响应数据【详见 签名生成规则 】
channelId	true	String	频道号
fileId	true	String	文件ID【可通过 查询频道录制视频信息 获取】

参数名	必选	类型	说明
deleteTimeFrame	true	String	要裁剪移除的时间区间 传入格式要求： (1) start、end值是单位为秒的数值，格式为： <code>[{"start":xx,"end":xx},{ "start":xx,"end":xx}]</code> (2) 多个区间请按照时间顺序从小到大排列 (3) 时间区间不允许重叠 (4) 时间区间不得超过100个限制 【示例：要裁剪移除的是第1s到第14s，以及第25s到第30s的区间，则传入的字符串为 <code>[{"start":1,"end":14},{ "start":25,"end":30}]</code>】 误差说明： (1) 裁剪区间存在一定的时间误差 (2) 误差原因：回放默认为m3u8格式裁剪，由于m3u8的视频裁剪是精确到每片ts，所以裁剪时会判断裁剪要移除的时间区间所包含的ts片，进行裁剪移除，如需更精准的裁剪，可联系保利威客服人员开通mp4裁剪
callbackUrl	false	String	裁剪成功或失败回调的url 【详见回调说明】
autoConvert	false	String	是否自动转存回放点到点播，默认为N Y：是 N：否
fileName	false	String	裁剪后文件名

示例

```
http://api.polyv.net/live/v3/channel/record/clip
```

表单参数：

```
fileName=3-25%E6%B5%8B%E8%AF%95%E8%A3%81%E5%89%AAN%E4%B8%8D%E8%BD%AC%E5%AD%98&appId=fr1r1zazn3&deleteTimeFrame=%5B%7B%22start%22%3A1%2C%22end%22%3A30%7D%2C%7B%22start%22%3A50%2C%22end%22%3A60%7D%5D&sign=BCA5136EEE29D14C0EA483BD5FC24464&autoConvert=N&callbackUrl=http%3A%2F%2Fxxxx&channelId=2191541&timestamp=1621841922358&fileId=249fd64bdf9329aa1e1c1228fb0732cd
```

响应参数描述

参数名	类型	说明
code	Integer	响应状态码，200为成功返回，非200为失败【详见 全局错误说明 】
status	String	响应状态文本信息
message	String	响应描述信息，当code为400或者500的时候，辅助描述错误原因
data	String	请求成功时返回提交成功信息 "submit success."：提交的视频正在裁剪处理中

回调说明

裁剪结果实时信息回调，可以及时通知用户裁剪任务的结果。

原理

通过HTTP接口（设置的callbackUrl）向用户服务器发送GET请求，将合并结果推送给用户服务器。

举例说明

举例如下：`http://abc.com/test.do?status=success&fileId=xxxx&channelId=xxx&sign=xxdxxxxx&clippedFileId=xxx&fileName=xxxx×tamp=xxx&actualTimeFrame=xxx`

参数名	类型	说明
status	String	接口处理结果，成功为success，失败为error
channelId	String	频道号，成功时返回
fileId	String	裁剪后的文件ID，成功时返回
clippedFileId	String	裁剪前的文件ID，成功时返回
fileUrl	String	裁剪后的视频地址，成功时返回
timestamp	String	回调的当前时间戳
sign	String	校验的加密字符串，生成的规则md5(AppSecret+timestamp)，AppSecret是直播系统的用密钥
actualTimeFrame	String	真实裁剪后保留的时间区间，经过urlEncode的字符串，格式为： <code>[{"start":xx, "end":xx}, {"start":xx, "end":xx}]</code>
fileName	String	文件名，经过urlEncode的字符串

Java请求示例

快速接入基础代码请[下载相关依赖源码](#)，[点击下载源代码](#)，下载后加入到自己的源码工程中即可。测试用例中的**HttpUtil.java** 和 **LiveSignUtil.java** 都包含在下载文件中。

强烈建议您使用[直播Java SDK](#)完成API的功能对接，直播Java SDK 对API调用逻辑、异常处理、数据签名、HTTP请求线程池进行了统一封装和优化。

```

private static final Logger log = LoggerFactory.getLogger(ChannelPlayBackTest.class);
/**
 * 异步裁剪直播录制文件
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void testClipRecordFile() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.appId;
    String appSecret = super.appSecret;
    String userId = super.userId;
    String timestamp = String.valueOf(System.currentTimeMillis());
    //业务参数
    String url = "https://api.polyv.net/live/v3/channel/record/clip";
    String channelId = "2191532";
    String fileId = "635b8e3c74d8a43530a138b0e75d841e";
    String deleteTimeFrame = "[{\"start\":1, \"end\":30},{\"start\":50, \"end\":60}]";
    String callbackUrl="http://xxx";
    String autoConvert="N";
    String fileName="fileName";

    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("channelId", channelId);
    requestMap.put("fileId", fileId);
    requestMap.put("deleteTimeFrame", deleteTimeFrame);
    requestMap.put("callbackUrl", callbackUrl);
    requestMap.put("autoConvert", autoConvert);
    requestMap.put("fileName", fileName);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    log.info("测试异步裁剪直播录制文件, 返回值: {}", response);
    //do somethings
}

```

响应示例

系统全局错误说明详见[全局错误说明](#)

成功示例

```

{
  "code":200,
  "status":"success",
  "message":"",
  "data":"submit success."
}

```

异常示例

```
{  
  "code": 400,  
  "status": "error",  
  "message": "invalid signature.",  
  "data": ""  
}
```