

合并直播录制文件并回调mp4下载地址

旧版接口地址[合并直播录制文件并回调mp4下载地址（旧版）](#)

接口描述

- 1、合并直播录制mp4文件，接口合并过程为异步处理过程

2、接口支持https协议

接口URL

http://api.polyv.net/live/v3/channel/record/merge-mp4-start

[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](#)
- 2、合并的录制文件开始与结束的间隔时间必须在8小时内
- 3、三分屏的录制将自动经过重制课件后再合并mp4
- 4、已合并返回文件地址（fileUrl） 30天内有效

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 】

参数名	必选	类型	说明
timestamp	true	Long	当前13位毫秒级时间戳，3分钟内有效
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
startTime	true	String	录制文件直播开始时间最小值，与endTime最大不能超过8小时，为13位毫秒级时间戳
endTime	true	String	录制文件直播开始时间最大值，与startTime最大不能超过8小时，为13位毫秒级时间戳
callbackUrl	false	String	合并成功或失败回调的url【详见 回调说明 】
fileName	false	String	合并后的视频的文件名，长度不能超过64

示例

```
http://api.polyv.net/live/v3/channel/record/merge-mp4-start
```

表单参数：

```
fileName=3-25%E6%B5%8B%E8%AF%95%E5%90%88%E5%B9%B6%E5%AF%BC%E5%87%BA%E5%90%8E%E6%96%87%E4%BB%B6%E5%90%8DMP4-1&appId=frlr1zazn3&sign=EE9979208A488FDBBFBBFFE381859C7F&startTime=1615452767000&endTime=1615461011000&channelId=2191541&timestamp=1621842028633
```

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	请求失败时为空，请求成功时为相关的信息【详见 data字段描述 】

data字段描述

参数名	类型	说明
fileId	String	文件ID
fileUrl	String	合并中返回空字符串，已合并成功返回MP4文件地址

回调说明

合并结果实时信息回调，可以及时通知用户合并任务的结果。

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×tamp=1603506240000&channelId=xxxx&fileIds=xxx,xxx&fileUrl=xxdxxxxx&fileName=urlxxxx`

参数名	类型	说明
status	String	接口处理结果，成功为success，失败为error

参数名	类型	说明
channelId	String	频道号
timestamp	Long	13位毫秒级时间戳
fileId	String	合并后的文件ID
fileIds	String	合并前的所有文件ID
fileUrl	String	合并后的MP4的地址
fileName	String	合并后的文件名称

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);
/**
 * 异步合并录制文件并回调MP4
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void testRecordMergeMp4() 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 = "http://api.polyv.net/live/v3/channel/record/merge-mp4";
    String channelId = "2191541";
    String startTime = "1615452025000";
    String endTime = "1615453127000";
    String callbackUrl = "http://abc.com";
    String fileName = "测试合并导出后文件名MP4";

    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("channelId", channelId);
    requestMap.put("startTime", startTime);
    requestMap.put("endTime", endTime);
    requestMap.put("callbackUrl", callbackUrl);
    requestMap.put("fileName", fileName);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.postFormBody(url, requestMap);
    log.info("测试异步合并录制文件并回调MP4, 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

1、提交的视频正在合并处理中

```
{
  "code":200,
  "status":"success",
  "message": "",
  "data":{
    "fileUrl": "",
    "fileId": "015748897d214bc0cad299c0ec43a706"
  }
}
```

2、合并文件已存在

```
{
  "code":200,
  "status":"success",
  "message": "",
  "data":{
    "fileUrl": "https://oss-live-1.videocc.net/record/record/recordf/1b348be3231615362355920a774/recordmerge-20210315161450-3ca1a3.mp4",
    "fileId": "015748897d214bc0cad299c0ec43a706"
  }
}
```

异常示例

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