

批量查询频道直播推流信息

接口描述

- 1、批量获取频道直播的实时推流信息

2、deployAddress、inAddress、lfr信息可能无法获取，返回值为null

3、接口支持https协议

接口URL

http://api.polyv.net/live/v3/channel/monitor/get-streams

在线API调用

请求方式

GET

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、获取频道直播流信息需要频道为直播中状态

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 】
channelIds	true	String	直播中的频道号，多个频道用英文逗号分隔，最多100个

示例

```
http://api.polyv.net/live/v3/channel/monitor/get-streams?
appId=frlr1zazn3&sign=6DDD5D27EAE6C9EE1713F9E2AEB59D99&channelIds=4184547%2C4184544%2C4146015%2C4146008&timestamp=1692322685465
```

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	Array	请求成功时为频道实时推流信息列表【详见 data字段描述 】

data字段描述

参数名	类型	说明
channelId	String	频道号

参数名	类型	说明
live	Boolean	频道是否直播中 true：直播中 false：未直播
streamInfo	Object	频道推流信息，直播中才有值，未直播时为null【详见 streamInfo字段描述 】

streamInfo字段描述

参数名	类型	说明
deployAddress	String	推送的CDN节点IP地址
inAddress	String	推流出口IP地址
streamName	String	流名
fps	String	推流帧率
lfr	String	推流丢帧率
inBandWidth	String	推流码率，单位：bps

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。

```

    /**
    * 批量查询频道直播流信息
    * @throws IOException
    * @throws NoSuchAlgorithmException
    */
@Test
public void getStreamsInfoTest() 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 timestamp = String.valueOf(System.currentTimeMillis());

    //业务参数
    String url = "http://api.polyv.net/live/v3/channel/monitor/get-streams";
    String channelIds = "4184547,4184544,4146015,4146008";

    //http 调用逻辑
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("channelIds", channelIds);

    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));

    String response = HttpUtil.get(url, requestMap);

    log.info("测试批量查询频道直播流信息成功: {}", 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": [
    {
      "channelId": "4146008",
      "live": false,
      "streamInfo": null
    },
    {
      "channelId": "4184547",
      "live": true,
      "streamInfo": {
        "deployAddress": "113.96.140.25",
        "inAddress": "106.55.200.246",
        "streamName": "1b448be32316919995568637549",
        "fps": "16",
        "lfr": "0.0",
        "inBandWidth": "64318"
      }
    },
    {
      "channelId": "4146015",
      "live": true,
      "streamInfo": {
        "deployAddress": null,
        "inAddress": "30.171.154.159",
        "streamName": "1b448be3231690474521694979d",
        "fps": "15",
        "lfr": null,
        "inBandWidth": "20756"
      }
    },
    {
      "channelId": "4184544",
      "live": true,
      "streamInfo": {
        "deployAddress": null,
        "inAddress": null,
        "streamName": "1b448be32316919995083869e11",
        "fps": "15.15",
        "lfr": null,
        "inBandWidth": "70058.266"
      }
    }
  ]
}

```

异常示例

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