

查询账号时间范围连麦使用量

接口描述

- 1、查询账号时间范围连麦使用量

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/statistics/mic/history/get

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、连麦使用量按天统计， 由于数据量和计算量大， 统计结果时效性为两天， 即今天的数据需后天才可查询（凌晨2点汇总）

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 】
startTime	false	Long	开始时间，13位毫秒级时间戳（仅支持到日期）
endTime	false	Long	结束时间，13位毫秒级时间戳（仅支持到日期）

示例

```
http://api.polyv.net/live/v4/statistics/mic/history/get?
appId=frlr1zazn3&sign=9DB0C793E311FD297EDCA9DBB9F0E3B4&startTime=1632672000000&endTime=1633017600000&timestamp=1657848439743
```

响应参数描述

参数名	类型	说明
code	Integer	状态码，与 http 状态码相同，用于确定基本的响应状态
status	String	响应结果，由业务决定，成功返回success，失败返回error
success	Boolean	是否成功响应
requestId	String	请求ID，每次请求生成的唯一的UUID，仅可用于排查、调试，不应该和业务挂上钩
error	Object	状态码非200时的错误信息【详见 Error字段描述 】

参数名	类型	说明
data	Object	【详见 data字段说明 】

Error参数描述

参数名	类型	说明
code	Integer	错误代码，用于确定具体的错误原因
desc	String	错误描述，与 error.code 对应

data参数描述

参数名	类型	说明
userId	String	用户ID
history	Integer	连麦使用分钟数

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(UserTest.class);
/**
 * 查询账号时间范围连麦使用量
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void testMicDuration() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际
    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/v4/statistics/mic/history/get";
    String startTime = "1632672000000";
    String endTime = "1633017600000";

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

    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.get(url, requestMap);
    log.info("测试查询账号时间范围连麦使用量, 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",
  "requestId": "2268eadf53ff49009a2c37927da52fd2.61.16578482803179325",
  "data": {
    "userId": "1b448be323",
    "history": 101
  },
  "success": true
}

```

异常示例

```
{
  "code": 400,
  "status": "error",
  "requestId": "d310b70bc329403f87f77f9203d50f89.128.16360831552223589",
  "error": {
    "code": 20001,
    "desc": "application not found."
  },
  "success": false
}
```