

获取观众列表

接口描述

- 1、分页获取观众列表

2、接口支持https协议

接口URL

https://api.polyv.net/live/v4/user/viewer-record/list

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](#)

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 】
source	false	String	来源（IMPORT:导入，WX:微信；MOBILE:手机）
mobile	false	String	手机号码
email	false	String	邮箱
area	false	String	地址
pageSize	false	String	页面大小（默认10，最大1000）
pageNumber	false	Integer	页码（目前限制最大值1000页）

请求示例

```
https://api.polyv.net/v4/user/viewer-record/list?
appId=frlr1zazn3&timestamp=1670471250000&sign=149397BFD607C518A841DAF8777D6A7E&mobile=demoData
```

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
status	String	响应结果，由业务决定，成功返回success，失败返回error

参数名	类型	说明
success	Boolean	响应结果，由业务决定，成功返回true，失败返回false
data	Object	分页用户信息【详见 Data字段说明 】
error	Object	状态码非200时的错误信息【详见 Error字段说明 】
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

Data参数描述

参数名	类型	说明
pageSize	Integer	页面大小
totalPages	String	剩余页码
totalItems	Boolean	剩余记录数
contents	Array	观众信息列表【详见 contents字段说明 】
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

Contents参数描述

参数名	必选	类型	说明
viewerUnionId	true	String	观众id
wxOpenId	false	String	微信openId
wxUnionId	false	String	微信UnionId
wxNickName	false	String	微信昵称
wxAvatar	false	String	微信头像

参数名	必选	类型	说明
wxAppld	false	String	微信Appld
mobile	false	String	手机号码
source	true	String	用户来源(IMPORT:导入)
name	false	String	采集姓名
lastCollectMobile	false	String	采集手机号码
email	false	String	邮箱
area	false	String	地址
latestAccessIp	false	String	最后访问IP
device	false	String	设备
watchDuration	false	Integer	观看时长
watchChannelCount	false	Integer	观看频道数
latestAuthTime	true	timestamp	最后授权时间
createTime	true	timestamp	创建时间

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

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 final Logger log = LoggerFactory.getLogger(getClass());
/**
 * 删除观众信息
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void getViewerTest() 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());
    String mobile = "demoData";
    //业务参数
    String url = "https://api.polyv.net/v4/user/viewer-record/list";

    //http 调用逻辑
    Map<String, String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp", timestamp);
    requestMap.put("viewerUnionId", viewerUnionId);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    String response = HttpUtil.get(url, requestMap);
    log.info("测试删除观众信息成功: {}", 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",
  "requestId": "5b33d00b-d284-4a3d-adcd-38210f8681c8",
  "data": {
    "pageSize": 5,
    "totalPages": 2,
    "totalItems": 10,
    "contents": [
      {
        "name": "demoData",
        "lastCollectMobile": "demoData",
        "email": "demoData",
        "area": "demoData",
        "latestAccessIp": "demoData",
        "device": "demoData",
        "watchDuration": 0,
        "watchChannelCount": 0,
        "viewerUnionId": "2_2v0u6u75hw013ja684ba9co7lnppq4j9",
        "mobile": "demoData9",
        "source": "IMPORT",
        "createTime": 1712114619000
      },
      {
        "name": "demoData",
        "lastCollectMobile": "demoData",
        "email": "demoData",
        "area": "demoData",
        "latestAccessIp": "demoData",
        "device": "demoData",
        "watchDuration": 0,
        "watchChannelCount": 0,
        "viewerUnionId": "2_x4r3er86b6h539o4clna3ss39xg3c169",
        "mobile": "demoData8",
        "source": "IMPORT",
        "createTime": 1712114608000
      },
      {
        "name": "demoData",
        "lastCollectMobile": "demoData",
        "email": "demoData",
        "area": "demoData",
        "latestAccessIp": "demoData",
        "device": "demoData",
        "watchDuration": 0,
        "watchChannelCount": 0,
        "viewerUnionId": "2_9965885141of3ec157yq69yf75xrt108",
        "mobile": "demoData7",
        "source": "IMPORT",
        "createTime": 1712114586000
      },
      {
        "name": "demoData",
        "lastCollectMobile": "demoData",
        "email": "demoData",
```

```

        "area": "demoData",
        "latestAccessIp": "demoData",
        "device": "demoData",
        "watchDuration": 0,
        "watchChannelCount": 0,
        "viewerUnionId": "2_2g0be0h9p6cd0777zq022yvze4a968bk",
        "mobile": "demoData6",
        "source": "IMPORT",
        "createTime": 1712114565000
    },
    {
        "name": "demoData",
        "lastCollectMobile": "demoData",
        "email": "demoData",
        "area": "demoData",
        "latestAccessIp": "demoData",
        "device": "demoData",
        "watchDuration": 0,
        "watchChannelCount": 0,
        "viewerUnionId": "2_y5t681288v0316n8z932e15d28w1ka2n",
        "mobile": "demoData5",
        "source": "IMPORT",
        "createTime": 1712114550000
    }
]
},
"success": true
}

```

异常示例

```

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

```