

分组添加观众

接口描述

- 1、分组添加观众

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/channel/lottery-viewer-list/create

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、若参数不正确则会响应400异常

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	Integer	频道号

请求体参数描述

参数名	必选	类型	说明
groupId	true	String	分组id
viewerIds	true	Array	观众id数组

示例

```
http://api.polyv.net/live/v4/channel/lottery-viewer-list/create?
appId=frlr1zazn3&timestamp=1665629374000&sign=6CFA3141718E1AF0155889EEA2988206&channelId=4619012
```

请求体json参数：

```
{
  "groupId": 677,
  "viewerIds": [
    "16",
    "17",
    "18"
  ]
}
```

响应描述

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 】
data	Array	响应详细信息【详见 data字段说明 】

Error响应描述

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响应描述

参数名	类型	说明
id	Long	主键id
groupId	String	分组id
viewerId	String	观众id

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());
/**
 * 分组添加观众
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void groupAdd() 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 url = "http://api.polyv.net/live/v4/channel/lottery-viewer-list/create";
    //业务参数
    Integer channelId = 5004544;
    Long groupId = 1L;
    List<String> viewerIds = Arrays.asList("16", "17", "18");

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

    Map<String, Object> jsonMap = new HashMap<>();
    jsonMap.put("groupId", groupId);
    jsonMap.put("viewerIds", viewerIds);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));

    url = HttpUtil.appendUrl(url, requestMap);
    String response = HttpUtil.postJsonBody(url, JSON.toJSONString(jsonMap), null);

    log.info("测试分组添加观众结果: {}", 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": "10b563c9-ce1f-4e5b-a959-64665af484bb",
  "data": [
    {
      "id": 439714,
      "groupId": 677,
      "viewerId": "16",
      "createTime": 1725518428600
    },
    {
      "id": 439715,
      "groupId": 677,
      "viewerId": "17",
      "createTime": 1725518428600
    },
    {
      "id": 439716,
      "groupId": 677,
      "viewerId": "18",
      "createTime": 1725518428600
    }
  ],
  "success": true
}
```

异常示例

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