

创建互动监听事件

接口描述

- 1、创建互动监听事件

2、接口支持https协议

接口URL

https://api.polyv.net/live/v4/channel/interaction-event/save

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、[任务列表](#)

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 】

请求体参数描述

参数名	必选	类型	说明
channelId	true	Integer	频道id
tasks	true	Array	任务规则列表【详见 tasks字段说明 】
callbackUrl	false	String	回调URL
allDone	true	String	是否tasks中的任务全部完成才算完成任务: Y 或 N
payload	false	String	有效负载

回调说明（POST请求）

参数名	类型	说明
roomId	String	房间ID
accountId	String	账号ID
content	String	用户完成任务列表，json数组，详见 content参数说明
requestId	String	请求ID
timestamp	Long	当前13位毫秒级时间戳

参数名	类型	说明
sign	String	签名验证，生成规则为： <code>polyvlog + 数据key和value组成 + polyvlog</code> ，其中 数据key 按首字母从小到大排序。md5转码后，转为大写。比如，参数为{roomId:'200060', channelId:'005200060',user:{nick:'yang'}}，则签名为 <code>\nmd5/('polyvChatSign'+channelId+'005200060'+roomId+'200060'+user'+JSON.stringify({nick:'yang'})+'polyvChatSign').toLocaleUpperCase();'</code>
totalBatch	Integer	当前任务总批次数
currentBatch	Integer	任务当前批次数

content 数组对象说明

参数名	类型	说明
data	Array	任务完成详情列表，详见 data列表说明
taskId	String	任务ID
activityId	String	活动ID
taskData	Object	任务自定义数据

data 列表说明

参数名	类型	说明
userId	String	用户ID

回调响应说明

回调接口的响应体需要满足以下任一条件，否则系统会进行重试，连续5次失败后放弃回调：

- `code` 字段的值为 `200`

- status 字段的值为 success

tasks参数描述

参数名	必选	类型	说明
startTime	true	Long	开始时间
endTime	true	Long	结束时间
type	true	String	任务条件(在线、签到)type为onlineTime时有onlineTime；type为signCount时有signCount；type为speakCount时有speakCount和speakContent，表示达到speakCount次，且评论内容为speakContent；type为customCount时有customCount；type为customCount时有eventType；type可以取值loginList；type为taskEndTimeOnline表示任务完成时，且用户在线，则触发任务
onlineTime	false	Integer	在线时长(单位为毫秒)
signCount	false	Integer	签到次数
userTags	false	Array	用户标签
speakCount	false	Integer	评论次数
speakContent	false	String	评论内容
customCount	false	Integer	自定义计数
eventType	false	String	事件类型
payload	false	String	有效负载，500字符以内

示例

```
https://api.polyv.net/live/v4/channel/interaction-event/save?
appId=frlr1zazn3&sign=012332FBD8DBCFC068AFCB484A2ADECB&timestamp=1670550787318
```

请求体json参数:

```
{
  "channelId": 5392252,
  "tasks": [
    {
      "type": "signCount",
      "signCount": 2,
      "startTime": 1733888246000,
      "endTime": 1733888426000
    }
  ],
  "callbackUrl" : "http://callback/write",
  "allDone": "Y"
}
```

响应体json:

```
{
  "code": 200,
  "status": "success",
  "requestId": "5b7d9087-c632-4607-93de-ed3da2c7a4cf",
  "data": {
    "activityId": "95564091-b835-11ef-9283-17521e995374",
    "list": [
      {
        "taskId": "95564090-b835-11ef-9283-17521e995374",
        "result": {
          "status": "success",
          "message": "",
          "code": 200,
          "data": ""
        }
      }
    ]
  },
  "message": "添加任务成功",
  "success": true
}
```

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
success	Boolean	响应结果，由业务决定，成功返回true，失败返回false
data	Object	里面返回taskId
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

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());
/**
 * 新建监听事件
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void createTest() 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 = "https://api.polyv.net/live/v4/channel/interaction-event/save";
    //业务参数
    Integer channelId = 5004544;
    String activityName = "demoData";
    Integer taskRule = 1;
    Long startTime = 1722035312584;
    Long endTime = 1722036312584;
    List<Task> tasks = new ArrayList();

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

    Map<String, Object> jsonMap = new HashMap<>();
    jsonMap.put("channelId", channelId);
    jsonMap.put("tasks", tasks);
    jsonMap.put("callbackUrl", "http://callback/write");
    jsonMap.put("allDone", "Y");

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

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

    log.info("测试新建监听事件: {}", 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": "5b7d9087-c632-4607-93de-ed3da2c7a4cf",
  "data": {
    "activityId": "95564091-b835-11ef-9283-17521e995374",
    "list": [
      {
        "taskId": "95564090-b835-11ef-9283-17521e995374",
        "result": {
          "status": "success",
          "message": "",
          "code": 200,
          "data": ""
        }
      }
    ]
  },
  "message": "添加任务成功",
  "success": true
}
```

异常示例

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

```
{
  "code": 200,
  "status": "success",
  "requestId": "29fc71ae-110f-43f2-87f0-4d60a6c743ff",
  "data": {
    "list": [
      {
        "taskId": "2368e870-b835-11ef-9283-17521e995374",
        "result": {
          "status": "error",
          "message": "time is too long",
          "code": 505,
          "data": {}
        }
      }
    ],
    "activityId": "2368e871-b835-11ef-9283-17521e995374"
  },
  "message": "添加任务成功",
  "success": true
}
```