

批量新增分组记录

接口描述

- 1、批量新增分组记录

2、分组记录数量上限是1万条

3、单次只允许增加同一频道的记录

4、校验参会人id是否重复时，不区分大小写

5、(channelId, timestamp, appId) 参与sign签名，并和sign一起通过url传递，请求体参数不参与签名，通过post请求体传递【请设置请求头contentType:application/json】

6、接口支持https协议

接口URL

https://api.polyv.net/live/v4/seminar/group/plan/save-batch

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](#)

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
list	true	Array	批量插入记录【详见 list字段说明 】

list字段说明

参数名	必选	类型	说明
viewerId	true	String	参会人id，长度不超过64位（不可重复）
nickname	true	String	参会人昵称，长度不超过64位（仅作为辨认依据，不改变用户参会名称）
groupNo	true	Integer	分组序号，1~50之间
groupRole	true	String	组内身份，一个分组内不能多于一个组长 leader：组长 member：组员

示例

```
https://api.polyv.net/live/v4/seminar/group/plan/save-batch?
appId=fyirmfdak4&timestamp=1657786379352&sign=CEA301164F9BA5076F156C1EB60F22F8&channelId=3220878
```

请求体参数

```
{
  "list" : [{
    "viewerId": "viewerId1",
    "nickname": "viewerId1_nick",
    "groupNo": 1,
    "groupRole": "leader"
  }, {

    "viewerId": "viewerId2",
    "nickname": "viewerId2_nick",
    "groupNo": 1,
    "groupRole": "member"
  }, {

    "viewerId": "viewerId3",
    "nickname": "viewerId3_nick",
    "groupNo": 2,
    "groupRole": "member"
  }]
}
```

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	失败原因，失败时返回。成功时空 【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字段说明

参数名	必选	类型	说明
infoIncorrectViewerIdList	Array	不合法的参会人id。（昵称为空/昵称长度大于64位/分组序号为空/分组序号不在1~50之间/组内身份为空/组内身份传参有误	
duplicateViewerIdList	Array	重复的参会人id	
duplicateLeaderGroupNoList	Array	重复组长的分组序号	

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(GroupPlanSaveBatch.class);
/**
 * 批量新增分组记录
 * @throws IOException
 */
@Test
public void testGroupPlanSaveBatch() 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 userId = super.userId;
    String timestamp=String.valueOf(System.currentTimeMillis());
    //业务参数
    String url = "https://api.polyv.net/live/v4/seminar/group/plan/save-batch";
    String channelId = "2191532";
    String body = "{\n" +
        "  \"list\" : [\n" +
        "    \"viewerId\": \"viewerId1\", \n" +
        "    \"nickname\": \"viewerId1_nick\", \n" +
        "    \"groupNo\": 1, \n" +
        "    \"groupRole\": \"leader\" \n" +
        "  ], \n" +
        "  \n" +
        "    \"viewerId\": \"viewerId2\", \n" +
        "    \"nickname\": \"viewerId2_nick\", \n" +
        "    \"groupNo\": 1, \n" +
        "    \"groupRole\": \"member\" \n" +
        "  ], \n" +
        "  \n" +
        "    \"viewerId\": \"viewerId3\", \n" +
        "    \"nickname\": \"viewerId3_nick\", \n" +
        "    \"groupNo\": 2, \n" +
        "    \"groupRole\": \"member\" \n" +
        "  ] \n" +
        "}";

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

    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    url = HttpUtil.appendUrl(url, requestMap);
    String response = HttpUtil.postJsonBody(url, body, null);
    log.info("测试批量新增分组记录接口返回值: {}", 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",
  "message": "",
  "data": true
}
```

异常示例

```
{
  "code": 400,
  "status": "error",
  "error": {
    "code": 10003,
    "desc": "时间戳过期"
  },
  "success": false
}
```

```
{
  "code": 403,
  "status": "error",
  "error": {
    "code": 10002,
    "desc": "签名错误"
  },
  "success": false
}
```

```
{
  "code": 400,
  "status": "error",
  "requestId": "f02d8d1c81b94bf3b761efb9b9bc1fd2.58.16577892258410007",
  "error": {
    "code": 50011,
    "desc": "批量插入分组记录失败, 信息有误"
  },
  "data": {
    "infoIncorrectViewerIdList": [],
    "duplicateViewerIdList": ["viewerid2", "viewerid3"],
    "duplicateLeaderGroupNoList": [],
    "correct": false
  },
  "success": false
}
```

```
{
  "code": 400,
  "status": "error",
  "requestId": "f02d8d1c81b94bf3b761efb9b9bc1fd2.58.16577892258410007",
  "error": {
    "code": 50010,
    "desc": "数量超过限制，最多10000条数据"
  },
  "data": {
    "infoIncorrectViewerIdList": [],
    "duplicateViewerIdList": [],
    "duplicateLeaderGroupNoList": [],
    "correct": false
  },
  "success": false
}
```