

创建集团分帐号

接口描述

- 1、创建集团分帐号
- 2、(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】
- 3、接口支持https协议

接口URL

http://api.polyv.net/live/v4/group/user/create

在线API调用

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、集团账号接口计算签名请使用集团主账号的appId、appSecret
- 3、分账号用量计费套餐跟随集团主账号, 分钟数计费只能分配分钟数, 并发计费只能分配并发数
- 4、expireDate账号过期时间不传或者格式错误, 默认跟随主账号
- 5、**账号创建与资源分配为异步操作**, 请调用本创建接口1~3s后再使用[查询分账号列表及剩余资源](#)查询账号信息

请求参数描述

参数名	必选	类型	说明
appId	true	String	集团主账号appId【详见 集团账号2.0获取密钥 】
timestamp	true	Long	当前13位毫秒级时间戳，3分钟内有效
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
email	true	String	邮箱
password	true	String	密码，需要同时包含数字、英文，长度8~32位
contacts	true	String	联系人
phone	true	String	手机号码，手机号不能绑定超过3个邮箱
maxChannels	true	Integer	最多可创建频道数量
minutes	false	Integer	分钟数，主账号为分钟数计费，该用量值必填
concurrent	false	Integer	并发数，主账号为并发计费，该用量值必填，不限制时设为-1
balance	false	Float	金额，不能小于0

参数名	必选	类型	说明
memo	false	String	备注，50字符长度
expireType	false	String	到期时间类型 group：跟随主账号 custom：自定义
expireDate	false	String	账号过期时间，格式 yyyyMMdd，如 20220428
remainFlow	false	Integer	点播流量，单位G，默认分 配100G
remainSpace	false	Integer	点播空间，单位G，默认分 配100G
aiPptVideoDuration	false	Integer	分配视频创作分钟数,不能 小于0
aiPptVideoWithDigitalHu manDuration	false	Integer	分配视频创作（含数字 人），不能小于 0
liveStatsCallbackUrl	false	String	直播统计数据回调地址， 需以http://或https://开头
aiPptVideoCallbackUrl	false	String	视频创作信息回调地址， 需以http://或https://开头

示例

```
http://api.polyv.net/live/v4/group/user/create?  
appId=g8dtv537aq&timestamp=1653485157000&sign=5F9F4CB3EEA92C7897418718973A2462
```

请求体参数

```
{
  "email": "polyvtest@polyv.net",
  "password": "****",
  "contacts": "polyvtest",
  "phone": 16666666666,
  "expireType": "custom",
  "expireDate": 20220520,
  "memo": "这是备注",
  "maxChannels": 100,
  "minutes": 100,
  "remainFlow": 100,
  "remainSpace": 100,
  "balance": 100,
  "liveStatsCallbackUrl": "https://www.abc.com/live-stats-callback",
  "aiPptVideoCallbackUrl": "https://www.abc.com/ai-ppt-video-callback"
}
```

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 】
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
appId	String	应用ID
appSecret	String	应用Secret
userId	String	用户全局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 static final Logger log = LoggerFactory.getLogger(getClass());
/**
 * 创建集团分帐号
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void createGroupUserTest() throws IOException, NoSuchAlgorithmException {
    //公共参数,填写自己的实际参数
    String appId = super.groupAppId;
    String appSecret = super.groupAppSecret;
    String timestamp = String.valueOf(System.currentTimeMillis());

    //业务参数
    String url = "http://api.polyv.net/live/v4/group/user/create";
    String email = "polyvtest123456@polyv.net";
    String password = "polyvtest123456";
    String contacts = "张三";
    String phone = "17600000000";
    String maxChannels = "10";
    String concurrent = "1000";
    String liveStatsCallbackUrl = "https://www.abc.com/live-stats-callback";
    String aiPptVideoCallbackUrl = "https://www.abc.com/ai-ppt-video-callback";

    //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("email", email);
    jsonMap.put("password", password);
    jsonMap.put("contacts", contacts);
    jsonMap.put("phone", phone);
    jsonMap.put("maxChannels", maxChannels);
    jsonMap.put("concurrent", concurrent);
    jsonMap.put("liveStatsCallbackUrl", liveStatsCallbackUrl);
    jsonMap.put("aiPptVideoCallbackUrl", aiPptVideoCallbackUrl);

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

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

    log.info("测试创建集团分帐号成功: {}", 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": "c7e83b9751fa4bfe9f57ba45494c147a.57.16529371726540445",
  "data": {
    "appId": "xxxxx",
    "appSecret": "xxxxxxxxxxxxx",
    "userId": "xxxxxx"
  },
  "success": true
}
```

异常示例

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