

全量保存用户等级列表

接口描述

- 1、全量保存账号的用户等级列表：请求中的等级列表会整体覆盖原有等级，未包含在请求中的等级将被删除，数组顺序即等级顺序
- 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、用户等级为账号级配置，账号下所有频道共用同一套等级；直播模板和频道只控制是否展示等级
- 4、管理后台【用户】→【用户等级】和开放API读写同一份配置，任一入口保存成功后，另一入口查询到相同结果
- 5、接口支持https协议

接口URL

http://api.polyv.net/live/v4/user/viewer-level/save

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、接口根据签名中的 `appId` 确定目标账号，不接收用户ID或频道号
- 3、每个账号至少保留1个、最多保存100个等级；同一请求中的等级标识 `levelId` 不能重复
- 4、已开通[保利威用户体系](#)的账号，每个等级都必须设置积分门槛 `requiredPoints`，且按数组顺序严格递增，即后一个等级的积分门槛必须大于前一个等级
- 5、未开通用户体系的账号只保存等级名称、图标、底色和说明等展示配置，请求中的 `requiredPoints` 不生效：已有等级（按 `levelId` 匹配）沿用原积分门槛，新增等级的积分门槛为0
- 6、建议先调用[查询用户等级列表](#)获取当前配置，在此基础上修改后整体提交，避免误删等级

7、用户等级的功能说明详见[用户等级](#)

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
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
levels	true	Array	等级列表，数组顺序即等级顺序，至少1个、最多100个【详见 levels参数描述 】

levels参数描述

参数名	必选	类型	说明
levelId	true	String	等级标识，账号内唯一；以字母或数字开头，仅支持字母、数字、下划线和中划线，最长64个字符
levelName	true	String	等级名称，展示给观众，最长10个字符

参数名	必选	类型	说明
levelIcon	false	String	等级图标URL，最长512个字符；未设置时展示等级名称和底色
levelBgColor	false	String	等级底色，格式为#RGB或#RRGGBB，如#FF6456
description	false	String	等级说明，仅用于后台识别和管理，不展示给观众，最长100个字符
requiredPoints	false	Long	积分门槛，即观众获得该等级所需的最低累计获得积分，取值1~999999 已开通用户体系时必须填，且按数组顺序严格递增 未开通用户体系时传入的值不生效

示例

```
http://api.polyv.net/live/v4/user/viewer-level/save?
appId=frlrlzazn3&sign=49428741C1D7BEE8BE1EE545F066ECC7&timestamp=1677167340447
```

请求体json参数：

```
{
  "levels": [
    {
      "levelId": "bronze",
      "levelName": "青铜",
      "levelIcon": "https://s1.videocc.net/default-img/level-icon/v1/bronze.png",
      "levelBgColor": "#FF6456",
      "description": "新用户默认等级",
      "requiredPoints": 1
    },
    {
      "levelId": "silver",
      "levelName": "白银",
      "levelBgColor": "#5E81FF",
      "requiredPoints": 1000
    }
  ]
}
```

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参数描述 】
data	Boolean	是否保存成功，true表示保存成功

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【详见 错误码说明 】
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 static final Logger log = LoggerFactory.getLogger(getClass());
/**
 * 全量保存用户等级列表
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void testViewerLevelSave() 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());

    //业务参数
    String url = "http://api.polyv.net/live/v4/user/viewer-level/save";
    List<Map<String, Object>> levels = new ArrayList<>();

    Map<String, Object> bronze = new HashMap<>();
    bronze.put("levelId", "bronze");
    bronze.put("levelName", "青铜");
    bronze.put("levelIcon", "https://sl.videocc.net/default-img/level-icon/v1/bronze.png");
    bronze.put("levelBgColor", "#FF6456");
    bronze.put("description", "新用户默认等级");
    bronze.put("requiredPoints", 1);
    levels.add(bronze);

    Map<String, Object> silver = new HashMap<>();
    silver.put("levelId", "silver");
    silver.put("levelName", "白银");
    silver.put("levelBgColor", "#5E81FF");
    silver.put("requiredPoints", 1000);
    levels.add(silver);

    //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("levels", levels);

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

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

    log.info("测试全量保存用户等级列表,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",
  "requestId": "414afeb28f1c4f0ca17e1c3c1e39d247.71.16771673423081591",
  "data": true,
  "success": true
}
```

异常示例

```
{
  "code": 400,
  "status": "error",
  "requestId": "d310b70bc329403f87f77f9203d50f89.128.16360831552223589",
  "error": {
    "code": 25105,
    "desc": "最低积分需大于上一等级的最低积分"
  },
  "success": false
}
```

错误码说明

错误码	错误描述	说明
10001	至少保留1个会员等级	未传levels，或levels为空数组
10001	最多支持100个会员等级	levels超过100个
10001	等级标识不能为空	未传levelId
10001	等级标识仅支持字母、数字、下划线和中划线，且以字母或数字开头	levelId格式不正确，或超过64个字符
10001	等级名称不能为空	未传levelName
10001	等级名称不能超过10个字符	levelName超过10个字符
10001	等级图标地址过长	levelIcon超过512个字符

错误码	错误描述	说明
10001	等级颜色格式不正确	levelBgColor不是#RGB或#RRGGBB格式
10001	等级描述不能超过100个字符	description超过100个字符
10001	最低积分不能小于1	requiredPoints小于1
10001	最低积分不能大于999999	requiredPoints大于999999
25102	等级标识不能重复	levels中存在相同的levelId
25104	最低积分必须为非负整数	已开通用户体系时，存在未传requiredPoints的等级
25105	最低积分需大于上一等级的最低积分	已开通用户体系时，requiredPoints未按数组顺序严格递增