

修改默认模板礼物打赏设置

接口描述

- 1、修改直播模板礼物打赏设置，礼物打赏又分为现金支付和积分支付

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/user/donate/gift/update

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](#)

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
donateGiftEnabled	true	String	礼物打赏开关，Y：开启，N：关闭
giftDonate	false	Object	礼物【详见 TemplateGiftDonate字段说明 】

TemplateGiftDonate参数描述

参数名	必选	类型	说明
payWay	false	String	支付方式，CASH：现金支付，POINT：积分支付
pointUnit	false	String	积分单位
cashPays	false	Array	现金支付列表，数组长度为1-18【详见 GiftDonate字段说明 】
pointPays	false	Array	积分支付列表，数组长度为1-16【详见 GiftDonate字段说明 】

GiftDonate参数描述

参数名	必选	类型	说明
name	false	String	礼物名称
img	true	String	礼物图片地址
price	false	Float	礼物价格，默认0.01
enabled	false	String	开关，Y：开启，N：关闭，默认为N

示例

```
http://api.polyv.net/live/v4/user/donate/gift/update?
appId=frlr1zazn3&sign=ECDDF7902925D558426182348FE3925D&timestamp=1631091439270
```

请求体json参数：

```
{
  "donateGiftEnabled": "Y",
  "giftDonate": {
    "cashPays": [
      {
        "img": "//s1.videocc.net/default-img/donate/666.png",
        "price": "666",
        "name": "现金礼物1",
        "enabled": "Y"
      }
    ],
    "pointUnit": "豆",
    "pointPays": [
      {
        "img": "//s1.videocc.net/default-img/donate/666.png",
        "price": "666",
        "name": "积分礼物1",
        "enabled": "Y"
      }
    ],
    "payWay": "CASH"
  }
}
```

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	成功响应时返回频道详细信息
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 static final Logger log = LoggerFactory.getLogger(UserDonateTest.class);

@Test
public void updateChannelDonate() throws IOException, NoSuchAlgorithmException {
    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/donate/gift/update";

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

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

    Map<String, Object> jsonMap = new HashMap<>();
    jsonMap.put("donateGiftEnabled", "Y");
    Map<String, Object> giftDonateMap = new HashMap<>();
    giftDonateMap.put("payWay", "CASH");
    giftDonateMap.put("pointUnit", "豆");
    List<Map<String, Object>> cashPaysList = new ArrayList<>();
    Map<String, Object> cash1 = new HashMap<>();
    cash1.put("name", "现金礼物1");
    cash1.put("img", "///s1.videocc.net/default-img/donate/666.png");
    cash1.put("price", "666");
    cash1.put("enabled", "Y");
    cashPaysList.add(cash1);
    giftDonateMap.put("cashPays", cashPaysList);
    List<Map<String, Object>> pointPaysList = new ArrayList<>();
    Map<String, Object> point1 = new HashMap<>();
    point1.put("name", "积分礼物1");
    point1.put("img", "///s1.videocc.net/default-img/donate/666.png");
    point1.put("price", "666");
    point1.put("enabled", "Y");
    pointPaysList.add(point1);
    giftDonateMap.put("pointPays", pointPaysList);
    jsonMap.put("giftDonate", giftDonateMap);

    url = HttpUtil.appendUrl(url, requestMap);

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

    log.info("测试修改直播模板礼物打赏设置, 返回值: {}", response);
}

```

响应示例

成功示例

```
{
  "code": 200,
  "status": "success",
  "requestId": "5d413aa5dc6b4cc1b79d4374aa5ac974.63.16310914532009701",
  "data": null,
  "success": true
}
```

异常示例

```
{
  "code": 400,
  "status": "error",
  "requestId": "4081dbac03e6441e8bdd301d8feee5a2.124.16360832547921601",
  "error": {
    "code": 20001,
    "desc": "application not found."
  },
  "success": false
}
```