

修改账号积分打赏设置

接口URL

http://api.polyv.net/live/v3/channel/donate/update-point

接口说明

- 1、接口用于设置账号通用设置的积分打赏
- 2、通用参数通过普通url参数传递（timestamp，appId，sign），设置打赏参数的json通过请求体的方式传递
- 3、接口支持https

支持格式

JSON

请求方式

POST

请求参数

带在url后面

参数名	必选	类型	说明
appId	是	string	从API设置中获取，在直播系统登记的appId
timestamp	是	long	13位当前时间戳

参数名	必选	类型	说明
sign	是	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 】

请求体传参

参数名	必选	类型	说明
donatePointEnabled	是	string	积分打赏开关，取值Y/N
queryPointUrl	是	string	获取观众积分打赏接口
updatePointUrl	是	string	修改观众积分打赏接口
requestFailTips	是	string	请求失败提示
pointNotEnoughTips	是	string	积分不足提示
pointUnit	是	string	积分单位
goods	是	object[]	积分打赏的道具列表
goodName	是	string	道具名称，不能超过5个字符
goodImg	是	string	道具图片，不能超过120个字符（图片地址可以使用默认地址，或者通过上传图片接口上传获取图片地址）
goodPrice	是	double	道具打赏价格
goodEnabled	是	string	道具开关，值为 Y/N，Y为开启

注：这里的channelId,appId,timestamp,sign必须通过url传参，json数据通过请求体传参，如：
<http://api.polyv.net/live/v3/channel/donate/update-point?appId={{appId}}×tamp=>

`{{timestamp}}&sign={{sign}}`

请求体json拼接

```
{
  "donatePointEnabled": "Y",
  "queryPointUrl": "xxx",
  "updatePointUrl": "yyy",
  "requestFailTips": "请求失败了",
  "pointNotEnoughTips": "积分不足啊",
  "pointUnit": "Q币",
  "goods": [
    {
      "goodName": "鲜花123",
      "goodImg": "//livestatic.videocc.net/uploaded/images/webapp/channel/donate/01-flower.png",
      "goodPrice": 123,
      "goodEnabled": "N"
    },
    {
      "goodName": "大跑车",
      "goodImg": "//livestatic.videocc.net/uploaded/images/webapp/channel/donate/08-car.png",
      "goodPrice": 999,
      "goodEnabled": "Y"
    }
  ]
}
```

响应成功JSON示例：

```
{
  "code": 200,
  "status": "success",
  "message": "",
  "data": ""
}
```

响应失败JSON示例：

#参数错误

```
{
  "code": 400,
  "status": "error",
  "message": "param validate error",
  "data": 400
}
```

#未输入appId

```
{
  "code": 400,
  "status": "error",
  "message": "appId is required.",
  "data": ""
}
```

#appId不正确

```
{
  "code": 400,
  "status": "error",
  "message": "application not found.",
  "data": ""
}
```

#时间戳错误

```
{
  "code": 400,
  "status": "error",
  "message": "invalid timestamp.",
  "data": ""
}
```

#签名错误

```
{
  "code": 403,
  "status": "error",
  "message": "invalid signature.",
  "data": ""
}
```

#json为空或者json解析错误

```
{
  "code": 400,
  "status": "error",
  "message": "param validate error",
  "data": ""
}
```

#json中的goods数量限制

```
{
  "code": 400,
  "status": "error",
  "message": "goods over limit",
  "data": ""
}
```

#json中的goods的名称字符限制

```
{
  "code": 400,
  "status": "error",
  "message": "goods name over limit",
  "data": ""
}
```

#json中的goods的图片地址字符限制

```
{
  "code": 400,
  "status": "error",
  "message": "goods image length over limit",
  "data": ""
}
```

#json中的goods的价格数据类型错误

```
{
  "code": 400,
  "status": "error",
  "message": "number range error",
  "data": ""
}
```

字段说明

参数名	说明
code	状态码,成功为200， 签名失败为403， 参数错误为400， 服务端错误为500

参数名	说明
status	成功为success,错误时为error
message	成功为"", 错误时为错误描述信息
data	暂无作用

java请求示例

```
import com.live.util.EncryptionUtils;
import org.apache.http.HttpEntity;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

import java.io.IOException;
import java.nio.charset.Charset;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class Demo {

    private static RequestConfig requestConfig =
RequestConfig.custom().setSocketTimeout(15000).setConnectTimeout(15000)
        .setConnectionRequestTimeout(15000).build();

    public static void main(String[] args) {

        String url = "http://api.polyv.net/live/v3/channel/donate/update-point";
        String appId = "XXXXXX";
        String key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
        Map<String, String> map = new HashMap<>();
        long timestamp = System.currentTimeMillis();
        map.put("appId", appId);
        map.put("timestamp", String.valueOf(timestamp));

        String sign = getSign(map, key);
        map.put("sign", sign);
        String body = "{\n\"goods\":[\n{\n\"goodName\":\n\"鲜花
333\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/01-
flower.png\", \"goodPrice\":0, \"goodEnabled\": \"Y\"}, {\n\"goodName\": \"咖啡
\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/02-
coffee.png\", \"goodPrice\":0.99, \"goodEnabled\": \"Y\"}, {\n\"goodName\": \"点赞
\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/03-
good.png\", \"goodPrice\":1.99, \"goodEnabled\": \"Y\"}, {\n\"goodName\": \"掌声
\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/04-
applaud.png\", \"goodPrice\":2.99, \"goodEnabled\": \"Y\"},
{\n\"goodName\": \"666\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/05-
666.png\", \"goodPrice\":6.66, \"goodEnabled\": \"Y\"}, {\n\"goodName\": \"小星星
\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/06-
star.png\", \"goodPrice\":9.99, \"goodEnabled\": \"Y\"}, {\n\"goodName\": \"钻石
\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/07-
diamond.png\", \"goodPrice\":19.99, \"goodEnabled\": \"Y\"}, {\n\"goodName\": \"跑车
\", \"goodImg\": \"//lifestatic.videocc.net/uploaded/images/webapp/channel/donate/08-
car.png\", \"goodPrice\":29.99, \"goodEnabled\": \"Y\"}, {\n\"goodName\": \"火箭
```

```

233\\",\\"goodImg\\":\\"//lvestatic.videocc.net/uploaded/images/webapp/channel/donate/09-
rocket.png\\",\\"goodPrice\\":66.66,\\"goodEnabled\\":\\"Y\\"}]]";
        String content = sendHttpPost(url, map, body);
        System.out.println(content);

    }

    /**
     * 发送 post请求
     * @param httpUrl 地址
     * @param maps 参数
     */
    public static String sendHttpPost(String httpUrl, Map<String, String> maps, String body) {
        StringBuilder url = new StringBuilder();
        url.append(httpUrl).append("?");
        for (Map.Entry<String, String> map : maps.entrySet()) {
            url.append(map.getKey()).append("=").append(map.getValue()).append("&");
        }
        String urlStr = url.toString().substring(0, url.length() - 1);
        // 创建httpPost
        HttpPost httpPost = new HttpPost(urlStr);
        try {
            StringEntity entity = new StringEntity(body, Charset.forName("UTF-8"));
            httpPost.setEntity(entity);
        } catch (Exception e) {
            // ...
        }
        return sendHttpPost(httpPost);
    }

    /**
     * 发送Post请求
     * @param httpPost
     * @return
     */
    private static String sendHttpPost(HttpPost httpPost) {
        CloseableHttpClient httpClient = null;
        CloseableHttpResponse response = null;
        HttpEntity entity;
        String responseContent = null;
        try {
            // 创建默认的httpClient实例.
            httpClient = HttpClients.createDefault();
            httpPost.setConfig(requestConfig);
            // 执行请求
            response = httpClient.execute(httpPost);
            entity = response.getEntity();
            responseContent = EntityUtils.toString(entity, "UTF-8");
        } catch (Exception e) {
            // ...
        } finally {
            try {
                // 关闭连接,释放资源
                if (response != null) {
                    response.close();
                }
            }
        }
    }

```



```

        if (null != httpPost) {
            httpPost.releaseConnection();
        }
        if (httpClient != null) {
            httpClient.close();
        }
    } catch (IOException e) {
        // ...
    }
}

return responseContent;
}

/**
 * 根据map里的参数构建加密串
 * @param map
 * @param secretKey
 * @return
 */
protected static String getSign(Map<String, String> map, String secretKey) {
    Map<String, String> params = paraFilter(map);
    // 处理参数, 计算MD5哈希值
    String concatedStr = concatParams(params);
    String plain = secretKey + concatedStr + secretKey;
    String encrypted = EncryptionUtils.md5Hex(plain);

    // 32位大写MD5值
    return encrypted.toUpperCase();
}

/**
 * 对params根据key来排序并且以key1=value1&key2=value2的形式拼接起来
 * @param params
 * @return
 */
private static String concatParams(Map<String, String> params) {
    List<String> keys = new ArrayList<>(params.keySet());
    Collections.sort(keys);

    StringBuilder sb = new StringBuilder();

    for (int i = 0; i < keys.size(); i++) {
        String key = keys.get(i);
        String value = params.get(key);

        sb.append(key).append(value);
    }

    return sb.toString();
}

/**
 * 除去数组中的空值和签名参数
 * @param sArray 签名参数组
 * @return 去掉空值与签名参数后的新签名参数组
 */

```

```
private static Map<String, String> paraFilter(Map<String, String> sArray) {

    Map<String, String> result = new HashMap<>();

    if (sArray == null || sArray.size() <= 0) {
        return result;
    }

    for (String key : sArray.keySet()) {
        String value = sArray.get(key);
        if (value == null || value.equals("") || key.equalsIgnoreCase("sign")
            || key.equalsIgnoreCase("sign_type")) {
            continue;
        }
        result.put(key, value);
    }

    return result;
}

}
```

php请求示例

```
<?php
//引用config.php
include 'config.php';

$params = array(
    'appId' => $appId,
    'timestamp' => $timestamp
);

//生成sign
$sign = getSign($params); //详细查看config.php文件的getSign方法

$json = '{"queryPointUrl": "xxx", "updatePointUrl": "yyy", "requestFailTips": "请求失败了", "pointNotEnoughTips":
"积分不足啊", "pointUnit": "Q币", "goods": [{ "goodName": "鲜花123", "goodImg":
 "//livestatic.videocc.net/uploaded/images/webapp/channel/donate/01-flower.png", "goodPrice": 123, "goodEnabled":
 "N"}, {"goodName": "大跑车", "goodImg": "//livestatic.videocc.net/uploaded/images/webapp/channel/donate/08-
car.png", "goodPrice": 999, "goodEnabled": "Y"}]}';
```