

# 修改全局回调设置

## 接口描述

1、修改全局回调设置

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/global-setting/callback/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](#)
- 2、提交的回调地址必须要以 http:// 或者 https:// 开头

## 请求参数描述

| 参数名       | 必选   | 类型     | 说明                                |
|-----------|------|--------|-----------------------------------|
| appId     | true | String | 账号appId【详见 <a href="#">获取密钥</a> 】 |
| 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 <a href="#">签名生成规则</a> 】 |

请求体参数描述

| 参数名                           | 必选    | 类型     | 说明                                                        |
|-------------------------------|-------|--------|-----------------------------------------------------------|
| recordCallbackUrl             | false | String | 录制生成回调URL                                                 |
| recordFileCallBackType        | false | String | 回调录制文件内容<br>all：全部回放视频<br>last：最终回放视频                     |
| recordCallbackVideoType       | false | String | 回调文件类型<br>m3u8：m3u8文件<br>mp4：mp4文件<br>m3u8,mp4：m3u8和mp4文件 |
| playbackCallbackUrl           | false | String | 转存成功回调URL                                                 |
| rebirthVodCallbackEnabled     | false | String | 重制课件转存点播回调开关<br>Y：开启<br>N：关闭                              |
| pptRecordCallbackUrl          | false | String | 课件重制成功回调URL                                               |
| streamCallbackUrl             | false | String | 直播状态改变回调URL                                               |
| channelBasicUpdateCallbackUrl | false | String | 频道直播间信息修改回调URL                                            |
| liveScanCallbackUrl           | false | String | 直播内容审核不通过回调URL                                            |

| 参数名                            | 必选    | 类型     | 说明              |
|--------------------------------|-------|--------|-----------------|
| playbackCacheCallbackUrl       | false | String | 直播回放缓存生成回调通知URL |
| aiPptVideoCallbackUrl          | false | String | 视频创作信息回调        |
| liveViolationCutoffCallbackUrl | false | String | 直播违规断流回调        |
| liveStatsCallbackUrl           | false | String | 直播数据统计信息回调      |

示例

```
http://api.polyv.net/live/v4/user/global-setting/callback/update?
appId=frlr1zazn3&sign=1C6F581887B59B388325338DA850330D&timestamp=1657852824050
```

请求体参数：

```
{
  "rebirthVodCallbackEnabled": "Y",
  "recordCallbackUrl": "https://www.abc.com/callback1",
  "streamCallbackUrl": "https://www.abc.com/callback3",
  "recordFileCallBackType": "last",
  "recordCallbackVideoType": "m3u8",
  "playbackCallbackUrl": "https://www.abc.com/callback2",
  "aiPptVideoCallbackUrl": "https://www.abc.com/callback4"
  "liveViolationCutoffCallbackUrl": "https://www.abc.com/callback4"
}
```

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                            |

| 参数名       | 类型      | 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     | Object  | 状态码非200时的错误信息【详见 <a href="#">Error字段描述</a> 】 |
| data      | Boolean | 成功返回true                                     |

### 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(UserTest.class);
/**
 * 修改全局回调设置
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void updateCallbackTest() 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/global-setting/callback/update";
    String recordCallbackUrl = "https://www.abc.com/callback1";
    String recordFileCallBackType = "last";
    String recordCallbackVideoType = "m3u8";
    String playbackCallbackUrl = "https://www.abc.com/callback2";
    String rebirthVodCallbackEnabled = "Y";
    String streamCallbackUrl = "https://www.abc.com/callback3";
    String aiPptVideoCallbackUrl = "https://www.abc.com/callback4";
    String liveViolationCutoffCallbackUrl = "https://www.abc.com/callback4";

    //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("recordCallbackUrl", recordCallbackUrl);
    jsonMap.put("recordFileCallBackType", recordFileCallBackType);
    jsonMap.put("recordCallbackVideoType", recordCallbackVideoType);
    jsonMap.put("playbackCallbackUrl", playbackCallbackUrl);
    jsonMap.put("rebirthVodCallbackEnabled", rebirthVodCallbackEnabled);
    jsonMap.put("streamCallbackUrl", streamCallbackUrl);
    jsonMap.put("aiPptVideoCallbackUrl", aiPptVideoCallbackUrl);
    jsonMap.put("liveViolationCutoffCallbackUrl", liveViolationCutoffCallbackUrl);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));

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

    log.info("测试修改全局回调设置成功: {}", 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": "8acd0194a7b6459a8ab168fcf2b63153.69.16578528273261669",
  "data": true,
  "success": true
}
```

## 异常示例

```
{
  "code": 400,
  "status": "error",
  "requestId": "d310b70bc329403f87f77f9203d50f89.128.16360831552223589",
  "error": {
    "code": 20001,
    "desc": "application not found."
  },
  "success": false
}
```