

更新用户身份信息

接口描述

- 1、更新用户身份信息

2、接口支持https协议

接口URL

https://api.polyv.net/live/v4/user/viewer-record/update

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
viewerUnionId	true	String	观众id
nickname	false	String	观众昵称，最大长度64个字符
mobile	false	String	手机号码，最大长度20个字符
name	true	String	收集姓名
lastCollectMobile	false	String	最后收集手机号码
email	true	String	邮箱
area	true	String	地址
latestAccessIp	true	String	最后访问IP
device	true	String	设备信息

示例

```
https://api.polyv.net/live/v4/user/viewer-record/update?
appId=fr1r1zazn3&sign=012332FBD8DBCFC068AFCB484A2ADECB&timestamp=1670550787318
```

请求体json参数：

```
{
  "viewerUnionId": "2_v378gn997yovt13p8h77db9e224t6hg9",
  "mobile": "3333",
  "nickname": "2222",
  "name": "demoData",
  "lastCollectMobile": "demoData",
  "email": "demoData",
  "area": "demoData",
  "latestAccessIp": "demoData",
  "device": "demoData"
}
```

响应体json:

```
{
  "code": 200,
  "status": "success",
  "requestId": "be18d7c8-4645-47f2-81dd-c3202d759b12",
  "data": true,
  "success": true
}
```

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	Boolean	响应新建成功的用户信息【详见 Data字段说明 】
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 final Logger log = LoggerFactory.getLogger(getClass());
/**
 * 更新观众信息
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void updateViewerTest() 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());

    String url = "https://api.polyv.net/live/v4/user/viewer-record/update";
    //业务参数
    String viewerUnionId = "2_v378gn997yovt13p8h77db9e224t6hg9";
    String mobile = "demoData";
    String nickname = "demoData";

    String name = "demoData";
    String lastCollectMobile = "demoData";
    String email = "demoData";
    String area = "demoData";
    String latestAccessIp = "demoData";
    String device = "demoData";

    //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("viewerUnionId", viewerUnionId);
    jsonMap.put("mobile", mobile);
    jsonMap.put("nickname", nickname);
    jsonMap.put("name", name);
    jsonMap.put("lastCollectMobile", lastCollectMobile);
    jsonMap.put("email", email);
    jsonMap.put("area", area);
    jsonMap.put("latestAccessIp", latestAccessIp);
    jsonMap.put("device", device);

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

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

    log.info("测试更新观众信息结果: {}", 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": "97d9bc9c-c0d2-4639-acad-54b28335004c",
  "data": {
    "name": "demoData",
    "lastCollectMobile": "demoData",
    "email": "demoData",
    "area": "demoData",
    "latestAccessIp": "demoData",
    "device": "demoData",
    "watchDuration": 0,
    "watchChannelCount": 0,
    "viewerUnionId": "2_2v0u6u75hw013ja684ba9co7lnppq4j9",
    "nickname": "demoData",
    "mobile": "demoData9",
    "source": "IMPORT",
    "latestAuthTime": 1712114620000,
    "createTime": 1712114620000
  },
  "success": true
}
```

异常示例

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