

修改轮询监控账号信息

接口描述

- 1、修改轮询监控账号信息
- 2、(channelId, 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

https://api.polyv.net/live/v4/seminar/monitor/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 】
channelId	true	Integer	频道号

请求体参数描述

参数名	必选	类型	说明
id	true	Integer	监控轮询账号
nickname	false	String	监播员昵称，支持1-16位长度（不传则不修改）
password	false	String	密码，支持6-16位长度（不传则不修改）

示例

```
https://api.polyv.net/live/v4/seminar/monitor/update?
appId=fyirmfdak4&timestamp=1657786379352&sign=CEA301164F9BA5076F156C1EB60F22F8&channelId=2191532
```

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

参数名	类型	说明
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字段描述 】
data	Object	失败原因，失败时返回。成功时为空【详见 data字段说明 】

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(UpdateAccountPost.class);
/**
 * 查询分组记录
 * @throws IOException
 */
@Test
public void testUpdateAccountPost() 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 userId = super.userId;
    String timestamp=String.valueOf(System.currentTimeMillis());
    //业务参数
    String url = "api.polyv.net/live/v4/seminar/monitor/update";
    String channelId = "2191532";
    String body = "{\n" +
        "    \"enable\": \"N\"\n" +
        "}";

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

    requestMap.put("sign",LiveSignUtil.getSign(requestMap, appSecret));
    url = HttpUtil.appendUrl(url, requestMap);
    String response = HttpUtil.postJsonBody(url, body, null);
    log.info("测试修改轮询监控账号信息接口返回值: {}",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": "faeae397448d4c5f988b278256cbc487.60.16591497134900021",
  "data": null,
  "success": true
}

```

异常示例

```
{
  "code": 400,
  "status": "error",
  "error": {
    "code": 10003,
    "desc": "时间戳过期"
  },
  "success": false
}
```

```
{
  "code": 403,
  "status": "error",
  "error": {
    "code": 10002,
    "desc": "签名错误"
  },
  "success": false
}
```

```
{
  "code": 400,
  "status": "error",
  "requestId": "faeae397448d4c5f988b278256cbc487.66.16591497085220019",
  "error": {
    "code": 1081,
    "desc": "没有监控轮询权限"
  },
  "data": null,
  "success": false
}
```

```
{
  "code": 400,
  "status": "error",
  "requestId": "b7463368632042daaf4d700631dda9e1.70.16593213159150141",
  "error": {
    "code": 50101,
    "desc": "监控轮询账号不存在"
  },
  "data": null,
  "success": false
}
```

```
{
  "code": 400,
  "status": "error",
  "requestId": "b7463368632042daaf4d700631dda9e1.69.16593219092970191",
  "error": {
    "code": 10001,
    "desc": "密码长度为6~16"
  },
  "data": null,
  "success": false
}
```

```
{
  "code": 400,
  "status": "error",
  "requestId": "b7463368632042daaf4d700631dda9e1.66.16593219248760193",
  "error": {
    "code": 10001,
    "desc": "昵称长度为1~16"
  },
  "data": null,
  "success": false
}
```