

设置重点直播列表

接口描述

- 1、给部分账号提供设置重点直播列表，仅开通功能的账号可以使用

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/channel/ccb/focus/reset

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 】
channelIds	false	String	重点直播的频道号列表，使用,分隔，会将原有设置的重点频道号进行覆盖，不传会清空重点直播

示例

```
http://api.polyv.net/live/v4/channel/ccb/focus/reset?
appId=ff0outsal5&sign=A6FF5B958C2D5103FB4A93334091ED93&channelIds=2731380%2C2750506%2C2737621%2C365032&timestamp=1642648261357
```

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
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
data	String	返回null
error	Object	错误信息【详见 Error参数描述 】

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。

```
@Test
public void testCCBFocusChannel() 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 channelId = "2731380,2750506,2737621,365032";
    String url = "http://api.polyv.net/live/v4/channel/ccb/focus/reset";

    //http 调用逻辑
    Map<String,String> requestMap = new HashMap<>();
    requestMap.put("appId", appId);
    requestMap.put("timestamp",timestamp);
    requestMap.put("channelIds", channelId);
    requestMap.put("sign", LiveSignUtil.getSign(requestMap, appSecret));
    url = MapUtil.appendUrl(url,requestMap);
    String response = HttpUtil.postFormBody(url, null);
    log.info("测试清空频道聊天记录接口返回值: {}",response);
}
```

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": "70fdf8c9808a4a799ffbdc3073630e3f.83.16426482643007221",
  "data": null,
  "success": true
}
```

异常示例

```
{
  "code": 400,
  "status": "error",
  "requestId": "4081dbac03e6441e8bdd301d8feee5a2.124.16360831818611581",
  "error": {
    "code": 20001,
    "desc": "application not found."
  },
  "success": false
}
```