

发送开播端弹窗消息

接口描述

- 1、用于直播中，给开播讲师以弹窗形式发送特定消息的能力（当前仅支持给讲师触发）

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/chat/alert-to-special

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](#)

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 】
channelId	true	String	频道号
message	true	String	消息说明 长度大于等于1，长度小于 等于500

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 】
timestamp	true	String	当前13位毫秒级时间戳，3分钟内有效
title	true	String	标题 长度大于等于1，长度小于等于100

示例

```
http://api.polyv.net/live/v4/chat/alert-to-special?
appId=frlr1zazn3&sign=4F16189E140DEAB547A5105093037D6B&message=%E6%88%91%E5%8F%91%E4%B8%AA%E5%BC%B9%E7%AA%97%E6%
B5%8B%E8%AF%95%E6%B6%88%E6%81%AF&title=%E8%AD%A6%E5%91%8A&channelId=4231598&timestamp=1693878611630
```

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
data	Boolean	成功响应的数据
error	Object	状态码非200时的错误信息【详见 Error参数描述 】
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
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

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(getClass());
/**
 * 提示特殊身份
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void chatAlertToSpecialTest() 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/chat/alert-to-special";
    String channelId = "4231598";
    String message = "我发个弹窗测试消息";
    String title = "警告";

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

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

    url = HttpUtil.appendUrl(url, requestMap);
    String response = HttpUtil.postFormBody(url, null);

    log.info("测试提示特殊身份成功: {}", 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": "33ffffbb646604f8a997724714b56ef1e.70.16938786176400585",
  "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
}
```