

发送审核通过的聊天消息

接口描述

- 1、发送审核通过的聊天消息
- 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

http://api.polyv.net/live/v4/chat/message/audit

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ID

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

请求体参数描述

参数名	必选	类型	说明
msgId	true	String	自定义的msgId，不超过32个字符
viewerId	true	String	观众id，不超过64个字符
nickName	true	String	观众昵称，不超过128个字符
content	true	String	消息内容content，不超过2000个字符
avatar	false	String	观众头像链接，不超过200个字符；若不传入该字段，viewerId对应观众在线时使用用户头像，否则使用默认头像
sessionId	false	String	场次id
viewerType	false	String	观众类型， viewer：观众， student: 学生， slice: 云课堂学员

示例

```
http://api.polyv.net/live/v4/chat/message/audit?
appId=fxlwyesvqd&sign=942AC99EACD1D11FF35E480414403EB0&timestamp=1680162971000&channelId=3808479
```

请求体json参数：

```
[
  {
    "msgId": "msg-1",
    "content": "我发送了一条消息",
    "viewerId": "sadboy",
    "nickName": "polyv"
  }
]
```

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	Array	成功响应的数据【详见 Data参数描述 】
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
success	Boolean	是否成功响应

Data参数描述

参数名	类型	说明
id	String	保利威聊天室消息id
msg	String	错误或正常的提示，如：发送成功
msgId	String	外部消息Id，即请求参数中的msgId
status	Boolean	发言是否成功，true：成功，false：失败

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());
/**
 * 审核通过聊天信息
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void messageAuditTest() 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/message/audit";
    String channelId = "1965681";

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

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

    url = HttpUtil.appendUrl(url, requestMap);

    Map<String, String> jsonMap = new HashMap<>();
    jsonMap.put("msgId", "msg-1");
    jsonMap.put("content", "我发送了一条消息");
    jsonMap.put("viewerId", "sadboy");
    jsonMap.put("nickName", "polyv");
    List<Map<String, String>> list = new ArrayList<>();
    list.add(jsonMap);

    String response = HttpUtil.postJsonBody(url, JSON.toJSONString(list), null);

    log.info("测试审核通过聊天信息成功: {}", 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": "807edf69deb540cd8b1f7a8b0e2dc883.64.16801629319610385",
  "data": [
    {
      "msgId": "123",
      "id": "3ec969a0-ced0-11ed-87bd-6de896f75816",
      "status": true,
      "msg": "发送成功"
    }
  ],
  "success": true
}
```

异常示例

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