

更新频道视频审核设置（需开通）

接口描述

- 1、更新频道视频审核设置
- 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/channel/video-moderation/update

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
channelId	true	Integer	频道id
moderationEnabled	true	String	审核开关，Y：开启，N：关闭
illegalNotify	true	Object	【详见 IllegalNotify参数描述 】
imageFrequency	true	Integer	截帧间隔（5：旗舰版(每分钟12审)；20：加强版(每分钟3审)；60：标准版(每分钟1审)）
moderationStrategy	true	String	审核策略， finance_easy：宽松审核策略， finance_normal：一般审核策略， finance_serious：严格审核策略

IllegalNotify参数描述

参数名	必选	类型	说明
assistantEnabled	false	String	助理通知，Y：开启，N：关闭，默认为N
monitorEnabled	false	String	监播通知，Y：开启，N：关闭，默认为N
platformEnabled	false	String	平台通知，Y：开启，N：关闭，默认为N
talentEnabled	false	String	主播通知，Y：开启，N：关闭，默认为N

示例

```
http://api.polyv.net/live/v4/channel/video-moderation/update?
appId=gga9dadhzm&sign=6AFE86666016B502E668174613B29585&channelId=3775168&timestamp=1679623766123
```

请求体json参数：

```
{
  "illegalNotify": {
    "platformEnabled": "Y",
    "monitorEnabled": "Y",
    "assistantEnabled": "Y",
    "talentEnabled": "Y"
  },
  "imageFrequency": "60",
  "moderationStrategy": "finance_serious",
  "moderationEnabled": "Y"
}
```

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

参数名	类型	说明
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

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());
/**
 * 更新频道视频审核设置
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void videoModerationUpdateTest() 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/channel/video-moderation/update";
    String channelId = "3775168";
    String moderationStrategy = "finance_serious";
    String imageFrequency = "60";
    String moderationEnabled = "Y";
    String assistantEnabled = "Y";
    String monitorEnabled = "Y";
    String platformEnabled = "Y";
    String talentEnabled = "Y";

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

    Map<String, Object> jsonMap = new HashMap<>();
    jsonMap.put("imageFrequency", imageFrequency);
    jsonMap.put("moderationEnabled", moderationEnabled);
    jsonMap.put("moderationStrategy", moderationStrategy);
    Map<String, Object> illegalNotifyMap = new HashMap<>();
    jsonMap.put("illegalNotify", illegalNotifyMap);
    illegalNotifyMap.put("assistantEnabled", assistantEnabled);
    illegalNotifyMap.put("monitorEnabled", monitorEnabled);
    illegalNotifyMap.put("platformEnabled", platformEnabled);
    illegalNotifyMap.put("talentEnabled", talentEnabled);

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

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

    log.info("测试更新频道视频审核设置成功: {}", 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": "f0103b0961754d3a887f82e35032a6ea.69.16796237824386651",
  "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
}
```