

置顶频道商品

接口描述

- 1、将指定频道商品置顶

2、(timestamp, appId, channelId) 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/product/topping

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](#)
- 2、仅允许对已存在且处于上架状态的商品执行置顶操作
- 3、同一频道仅允许一个商品处于置顶状态, 置顶新商品会自动取消原置顶商品

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
productId	true	Integer	频道商品ID

示例

```
http://api.polyv.net/live/v4/channel/product/topping?
appId=fr1r1zazn3&timestamp=1710468881000&sign=2F87D9D7F1234567890ABCDEFFEDCBA&channelId=1965681
```

请求体json参数：

```
{
  "productId": 27537
}
```

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
data	Object	响应数据，成功时通常为null
error	Object	状态码非200时的错误信息【详见 Error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(ChannelProductTest.class);
/**
 * 置顶频道商品
 */
@Test
public void testToppingChannelProduct() 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/product/topping";
    String channelId = "1965681";
    Integer productId = 27537;

    //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);

    Map<String, Object> bodyMap = new HashMap<>();
    bodyMap.put("productId", productId);

    String response = HttpUtil.postJsonBody(url, JSON.toJSONString(bodyMap), null);
    log.info("测试置顶频道商品, 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": "ac58000e9651475aaf4f48d7dae318cd.72.17104688816896941",
  "data": null,
  "success": true
}

```

异常示例（商品不存在）

```
{
  "code": 400,
  "status": "error",
  "requestId": "d310b70bc329403f87f77f9203d50f89.128.16360831552223589",
  "error": {
    "code": 40243,
    "desc": "商品不存在"
  },
  "success": false
}
```

异常示例（下架商品不允许操作）

```
{
  "code": 400,
  "status": "error",
  "requestId": "f92e5d4f3fd948c08c46c64d6f2329f7.57.17104689999231312",
  "error": {
    "code": 40258,
    "desc": "下架商品不允许操作"
  },
  "success": false
}
```