

获取频道商品统计概览

接口描述

- 1、查询频道商品整体统计数据概览（点击、下单、成交等指标汇总）

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/channel/product/stats/summary

在线API调用

请求方式

GET

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、若参数不正确则会响应400异常

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
sessionId	false	String	场次ID

示例

```
http://api.polyv.net/live/v4/channel/product/stats/summary?
appId=frlr1zazn3&timestamp=1665629374000&sign=6CFA3141718E1AF0155889EEA2988206&channelId=2974342
```

响应描述

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
error	Object	状态码非200时的错误信息【 详见Error响应描述 】
data	Object	商品统计概览数据【 详见data响应描述 】

参数名	类型	说明
encryption	Boolean	数据是否加密

Error响应描述

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

data响应描述

参数名	类型	说明
payPageNum	Integer	下单按钮点击次数
payPageViewers	Integer	下单按钮点击人数
payViewers	Integer	成交人数
productClickNum	Integer	商品点击次数
productClickViewers	Integer	商品点击人数
saleQuantity	Integer	商品销量
totalPayAmount	Number	成交金额（单位：元）

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(ChannelOperateTest.class);
/**
 * 查询频道商品统计概览
 * @throws IOException
 * @throws NoSuchAlgorithmException
 */
@Test
public void testGetChannelProductStatsSummary() 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/stats/summary";
    String channelId = "2779045";

    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));
    String response = HttpUtil.get(url, requestMap);
    log.info("测试查询频道商品统计概览, 返回值: {}", 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": "f46e79d6c5a94dcc9a40de76a8ffa777.74.17306917836670585",
  "data": {
    "payPageNum": 0,
    "payPageViewers": 0,
    "payViewers": 0,
    "productClickNum": 0,
    "productClickViewers": 0,
    "saleQuantity": 0,
    "totalPayAmount": 0
  },
  "encryption": true,
  "success": true
}

```

异常示例

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