

查询商品订单详情

接口描述

- 1、查询商品订单详情

2、接口支持https协议

接口URL

http://api.polyv.net/live/v4/user/product/order/get

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 】
orderNo	true	String	订单ID

示例

```
http://api.polyv.net/live/v4/user/product/order/get?
appId=fr1r1zazn3&timestamp=1665629374000&sign=6CFA3141718E1AF0155889EEA2988206&pageNumber=1
```

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字段描述 】
data	Object	响应详细信息【详见 ProductOrder字段说明 】

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

ProductOrder响应描述

参数名	类型	说明
orderNo	String	订单ID
channelId	Integer	频道ID
viewerId	String	下单用户ID
viewerNickName	String	下单用户昵称
transactionId	String	微信订单号,支付成功才有
amount	BigDecimal	订单金额
expressEnabled	String	是否有物流，Y是、N否
status	String	订单状态，待支付:wait_pay、待发货:wait_delivery、待收货:delivering、已完成:finish、已关闭:close
createTime	Long	创建时间,13位毫秒级时间戳
payTime	Long	支付时间,13位毫秒级时间戳
deliveryTime	Long	发货时间,13位毫秒级时间戳
finishTime	Long	完成时间,13位毫秒级时间戳
addressExt	Object	收货地址信息【详见 AddressExt字段说明 】
products	Array	订单商品信息【详见 Products字段说明 】

AddressExt响应描述

参数名	类型	说明
mobile	String	手机号
name	String	收货人姓名
area	String	收货地址省市信息
address	String	收货地址详细地址

Products响应描述

参数名	类型	说明
productSnapshotId	Long	商品快照ID
amount	BigDecimal	商品单价
quantity	Integer	商品数量
totalAmount	BigDecimal	商品总金额
productName	String	商品名称
productCover	String	商品封面图片

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(ProductOrderTest.class);  
/**  
 * 查询商品订单  
 * @throws IOException  
 */  
@Test  
public void testGetProductOrder() 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/user/product/order/get";  
    String orderNo = "";  
    //http 调用逻辑  
    Map<String,String> requestMap = new HashMap<>();  
    requestMap.put("appId", appId);  
    requestMap.put("timestamp",timestamp);  
    requestMap.put("orderNo",orderNo);  
    requestMap.put("sign",LiveSignUtil.getSign(requestMap, appSecret));  
    String response = HttpUtil.get(url, requestMap);  
    log.info("测试查询商品订单: {}",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": "ffec4cf4227c42d09f780ed0d9aa19e7.71.17210379136070035",
  "data": {
    "orderNo": "product202405221129011833ad",
    "channelId": 4872391,
    "viewerId": "c_4nv36e832udqged0g7t565t4zoasuo9y",
    "viewerNickName": "斌",
    "transactionId": "4200002204202405229969120950",
    "amount": 0.01,
    "status": "finish",
    "expireTime": 1716349441000,
    "payTime": 1716348582000,
    "deliveryTime": 1717746827000,
    "finishTime": 1717747300000,
    "createTime": 1716348541000,
    "addressExt": {
      "mobile": "15812345678",
      "name": "小明",
      "area": "广东省/广州市/天河区",
      "address": "详细地址01"
    },
    "products": [
      {
        "productSnapshotId": 1,
        "amount": 0.01,
        "quantity": 1,
        "totalAmount": 0.01,
        "productCover": "//liveimages.videocc.net/uploaded/images/2024/05/gwc3u3jkh0.jpg",
        "productName": "键盘",
        "deliveryType": "express",
        "customerServiceType": null,
        "customerServiceContact": null
      }
    ],
    "expressEnabled": "Y"
  },
  "success": true
}
```

异常示例

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