# logo_v21xZhiq1you

**海带开放平台API接口文档**

V3.0.9

**文档编号：**HDAPI\_2019\_0515\_V3

www.seatent.com

**日期：**2019-5-15

**更改记录**

| 日期 | 作者 | 版本 | 更改参考 |
| --- | --- | --- | --- |
| 2017-09-14 | 无缺 | V2.0 | 创建 |
| 2017-11-15 | 空明 | V2.0 | 修改 |
| 2017-11-28 | 无缺 | V2.1 | 新增资金流水接口 |
| 2018-04-25 | 无缺 | V3.0 | 出入参变动接口： 登录接口 地区接口创建订单取消申请售后售后订单列表申请售后确认收货获取订单列表获取订单详情 |
| 2018-05-07 | 无缺 | V3.0.1 | 修改地区json |
| 2018-05-14 | 无缺 | V3.0.2 | 订单列表接口添加入参：第三方订单号（customOrder） |
| 2018-05-15 | 无缺 | V3.0.3 | 更正登录相关地址更正json相关地址，修改登录接口注意事项。 |
| 2018-07-12 | 空明 | V3.0.4 | 修改登录接口注意事项。修改地区json地址 |
| 2018-08-7 | 贝尔 | V3.0.5 | 获取地区Json修改去掉入参，示范json调整 |
| 2019-01-10 | 杨过 | V3.0.6 | 创建订单新增，保税商品下单buyAmount参数必填 |
| 2019-01-24 | 杨过 | V3.0.7 | 新增“获取保税订单支付二维码”API |
| 2019-02-28 | 杨过 | V3.0.8 | 修正创建订单代购价参数名为thirdBuyAmount |
| 2019-05-15 | 杨过 | V3.0.9 | 新增获取支付方式、发起支付、查询支付状态接口 |
| 2019-12-27 | 子龙 | V3.1.0 | 新增保存支付信息 |

**目 录**

[第一章 前言 9](#_Toc8891339)

[1.1业务背景 9](#_Toc8891340)

[1.2商品接口调用流程图 9](#_Toc8891341)

[1.3订单接口调用流程图 10](#_Toc8891342)

[1.4订单查询流程图（正常及异常订单） 10](#_Toc8891343)

[1.5售后流程图 11](#_Toc8891344)

[第二章 接口授权 11](#_Toc8891345)

[2.1 授权说明 11](#_Toc8891346)

[2.2 接口请求地址 12](#_Toc8891347)

[2.3 签名规则 12](#_Toc8891348)

[第三章 核心接口 13](#_Toc8891349)

[3.1 接口登录 13](#_Toc8891350)

[3.1.1 应用场景 13](#_Toc8891351)

[3.1.2 注意事项 13](#_Toc8891352)

[3.1.3 API内容 14](#_Toc8891353)

[3.1.4 返回json串 示例 16](#_Toc8891354)

[3.2 商品类目 16](#_Toc8891355)

[3.2.1 应用场景 16](#_Toc8891356)

[3.2.2 注意事项 16](#_Toc8891357)

[3.2.3 API内容 16](#_Toc8891358)

[3.2.4 返回json串 示例 18](#_Toc8891359)

[3.3 商品列表 19](#_Toc8891360)

[3.3.1 应用场景 19](#_Toc8891361)

[3.3.2 注意事项 19](#_Toc8891362)

[3.3.3 API内容 19](#_Toc8891363)

[3.3.4 返回json串 示例 26](#_Toc8891364)

[3.4 获取商品详情 28](#_Toc8891365)

[3.4.1 应用场景 28](#_Toc8891366)

[3.4.2 注意事项 29](#_Toc8891367)

[3.4.3 API内容 29](#_Toc8891368)

[3.4.3 返回json串 示例 36](#_Toc8891369)

[3.5 获取商品动态价格 39](#_Toc8891370)

[3.5.1 应用场景 39](#_Toc8891371)

[3.5.2 注意事项 39](#_Toc8891372)

[3.5.3 API内容 39](#_Toc8891373)

[3.5.4 返回json串 示例 42](#_Toc8891374)

[3.6 商品多规格信息 43](#_Toc8891375)

[3.6.1 应用场景 43](#_Toc8891376)

[3.6.2 注意事项 43](#_Toc8891377)

[3.6.3API内容 44](#_Toc8891378)

[3.5.4 返回json串 示例 45](#_Toc8891379)

[3.7 获取运费 46](#_Toc8891380)

[3.7.1 应用场景 46](#_Toc8891381)

[3.7.2 注意事项 46](#_Toc8891382)

[3.7.2 API内容 46](#_Toc8891383)

[3.7.2 返回json串 示例 48](#_Toc8891384)

[3.8 创建订单 48](#_Toc8891385)

[3.8.1 应用场景 48](#_Toc8891386)

[3.8.2 注意事项 48](#_Toc8891387)

[3.8.3 API内容 49](#_Toc8891388)

[3.8.4 返回json串 示例 51](#_Toc8891389)

[3.9 获取订单列表 52](#_Toc8891390)

[3.9.1 应用场景 52](#_Toc8891391)

[3.9.2 注意事项 53](#_Toc8891392)

[3.9.3 API内容 53](#_Toc8891393)

[3.9.4 返回json串 示例 59](#_Toc8891394)

[4.0 获取订单详情 61](#_Toc8891395)

[4.0.1 应用场景 61](#_Toc8891396)

[4.0.2 注意事项 61](#_Toc8891397)

[4.0.3 API内容 62](#_Toc8891398)

[4.0.4 返回json串 示例 68](#_Toc8891399)

[4.1 获取地区信息 71](#_Toc8891400)

[4.1.1 应用场景 71](#_Toc8891401)

[4.1.2 注意事项 71](#_Toc8891402)

[4.1.3 API内容 71](#_Toc8891403)

[4.1.4 返回json串 示例 71](#_Toc8891404)

[4.2 修改报关信息接口 74](#_Toc8891405)

[4.2.1 应用场景 74](#_Toc8891406)

[4.2.2 注意事项 75](#_Toc8891407)

[4.2.3 API内容 75](#_Toc8891408)

[4.2.4 返回json串 示例 76](#_Toc8891409)

[4.3 申请售后接口 76](#_Toc8891410)

[4.3.1 应用场景 76](#_Toc8891411)

[4.3.2 API内容 76](#_Toc8891412)

[4.3.3 返回json串 示例 77](#_Toc8891413)

[4.4 确认收货接口 78](#_Toc8891414)

[4.4.1 应用场景 78](#_Toc8891415)

[4.4.2 API内容 78](#_Toc8891416)

[4.4.3 返回json串 示例 79](#_Toc8891417)

[4.5 获取快递信息接口 79](#_Toc8891418)

[4.5.1 应用场景 79](#_Toc8891419)

[4.5.2 注意事项 79](#_Toc8891420)

[4.5.3 API内容 79](#_Toc8891421)

[4.5.4 返回json串 示例 81](#_Toc8891422)

[4.6 取消订单 82](#_Toc8891423)

[4.6.1 应用场景 82](#_Toc8891424)

[4.6.2 注意事项 82](#_Toc8891425)

[4.6.3 API内容 82](#_Toc8891426)

[4.6.4 返回json串 示例 83](#_Toc8891427)

[4.7 取消申请售后接口 84](#_Toc8891428)

[4.7.1 应用场景 84](#_Toc8891429)

[4.7.2 API内容 84](#_Toc8891430)

[4.7.3 返回json串 示例 85](#_Toc8891431)

[4.8 发起余额扣款接口 85](#_Toc8891432)

[4.8.1 应用场景 85](#_Toc8891433)

[4.8.2 注意事项 85](#_Toc8891434)

[4.8.3 API内容 85](#_Toc8891435)

[4.8.4 返回json串 示例 86](#_Toc8891436)

[4.9 交易资金流水列表 87](#_Toc8891437)

[4.9.1 应用场景 87](#_Toc8891438)

[4.9.2 API内容 87](#_Toc8891439)

[4.9.3 返回json串 示例 89](#_Toc8891440)

[5.0 获取保税订单支付二维码 90](#_Toc8891441)

[5.0.1 应用场景 90](#_Toc8891442)

[5.0.2 API内容 90](#_Toc8891443)

[5.0.3 返回json串 示例 91](#_Toc8891444)

[5.1获取支付方式列表 91](#_Toc8891445)

[5.1.1 应用场景 91](#_Toc8891446)

[5.1.2 API内容 91](#_Toc8891447)

[5.1.3 返回json串 示例 92](#_Toc8891448)

[5.2发起支付 93](#_Toc8891449)

[5.2.1 应用场景 93](#_Toc8891450)

[5.2.2 API内容 93](#_Toc8891451)

[5.2.3 返回json串 示例 94](#_Toc8891452)

[5.3查询支付结果 95](#_Toc8891453)

[5.3.1 应用场景 95](#_Toc8891454)

[5.3.2 API内容 95](#_Toc8891455)

[5.3.3 返回json串 示例 96](#_Toc8891456)

# 第一章 前言

## 1.1业务背景

本文档为第三方电商平台与海带开放平台API对接技术实现方案。其中主要内容为技术接口说明，包括了商品类目、商品、订单、相关接口技术说明。

## 1.2商品接口调用流程图



## 1.3订单接口调用流程图



## 1.4订单查询流程图（正常及异常订单）



## 1.5售后流程图



# 接口授权

## 2.1 授权说明

为保证第三方平台与海带开放平台间接口安全，防止外部接口滥用，需设定APP\_KEY，APP\_SECRET，开发时需先获取测试服务器地址及相应的APP\_KEY和APP\_SECRECT。

通过APP\_KEY和APP\_SECRET进行用户登录，登录后获取与APP\_KEY关联用户的member\_id，account\_id和token后再进行后续接口调用，并叠加member\_id,accountId和token参数（接口定义不再复述）。

开放平台通用接口返回参数中的result定义如下：

0 功能调用失败

1 功能调用成功

开放平台通用接口返回参数中的code定义如下

101 密钥不正确（可能原因：APP\_SECRECT不正确）

102 appkey不正确（解决办法：检查是否拼写错误，参数传递顺序是否正确）

103 请求超时（解决办法：重新请求）

104 时间戳错误（可能原因：参数timestamp错误；一定要是北京时间；时间戳13 位）

105 签名验证失败（可能原因：url中存在中文或参数值有标点符号的需要url encode utf-8转码）

106 授权认证失败（可能原因：查看memberId参数，token参数是否正常）

## 2.2 接口请求地址

测试环境：http://api.demo.saaspre.seatent.com

#### 测试环境

appkey =86065876

secret = 72725d7bb8f04f839ffb19fae0d7851e

memberId = d3550dbaa993409ea1b222c5aa00c5ab

accountId = d3550dbaa993409ea1b222c5aa00c5ab

账号 = 17800000623

密码 = 123456

例：登录接口：

http://api.demo.saaspre.seatent.com/ssoapi/v2/login/login

## 2.3 签名规则

把 URL 请求的所有参数，除了 topSign 本身以外。把所有的参数按照字母大小排列（从小到大，同样字母则以第二个字母开始排序）。

示例：

接口登录

/ssoapi/v2/login/login?appkey=93028089&password=e10adc3949ba59abbe56e057f20f883e&timestamp=1646665656007&username=13700015820&topSign=27D859934ED04A57684A3B919EE7CBCB4175C84F

签名过程：

第一步：字符串排序

appkey=93028089&password=e10adc3949ba59abbe56e057f20f883e&timestamp=1646665656007&username=13700015820

第二步：第一步字符串前后增加 APP\_SECRET，假设 APP\_SECRET 是 2024ffe2897140ef989988e957007c1c

2024ffe2897140ef989988e957007c1cappkey=93028089&password=e10adc3949ba59abbe56e057f20f883e&timestamp=1646665656007&username=137000158202024ffe2897140ef989988e957007c1c

（注意：有中文或标点符号请先encode utf-8转码）

第三步：对第二步字符串做 SHA1 加密(大写，请以实际签名结果为准)

27D859934ED04A57684A3B919EE7CBCB4175C84F

请求的时候，带上 topSign=27D859934ED04A57684A3B919EE7CBCB4175C84F即可

注意：timestamp 的长度为 13 位且务必为准确时间

# 第三章 核心接口

## 3.1 接口登录

### 3.1.1 应用场景

需要获取商品列表时，需要会员id。

### 3.1.2 注意事项

1. 禁止多线程调用；
2. 多线程请加锁；
3. 全站共享token，禁止多线程并发调用；
4. 修改密码后需要重新登录，接口返回错误码106的时候需要重新调用登录接口。
5. 登录成功后，请将 accountId，memberId，token，appkey，timestamp, 签名后的topSign作为必传项在后续的接口调用中。
6. 获取json文件时，需要将登录接口返回的siteId作为路径参数如： [http://img.pre.seatent.com](http://api.pre.seatent.com)/statics/json/{siteId}/abc.png
7. 获取地址json ：

新地址json：[http://img.pre.seatent.com](http://api.pre.seatent.com)/statics/json/regionsAll.json

新老地址id对照json：[http://img.pre.seatent.com](http://api.pre.seatent.com)/statics/json/oldRegions.json

### 3.1.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /ssoapi/v2/login/login |
| 功能描述 | 登录获取token 、member\_id、accountId等 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
|  | 用户名 | string | 必填；开放平台用户名 |
| password | 密码 | string | 必填；开放平台密码（需将明文进行MD5加密后传输） |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code | 错误码 | string | @mock="" |
| success |  | boolean | @mock=true |
| result | 结果（1:成功;0:失败） | number | @mock=1 |
| data |  | object |  |
| siteId | ---站点ID | string | 获取json时所用的siteId |
| userName | ------用户名称 | string | @mock=15858121765 |
| mobile | ------手机号 | string | @mock=15858121765 |
| memberId | ------用户编号 | string | @mock= bf3ea2428fa73434068c661106980b70ff4c8fd70 |
| accountId | -----账户编号 | string | @mock= bf3ea2428fa73434068c661106980b70ff4c8fd70 |
| lvId | ------等级编号 | number | @mock=2 |
| lvName | ------等级名称 | string | @mock=Vip2 |
| facePic | ------头像图片 | string | @mock=http://seatent.com/1.jpg |
| token | ---token | string | @mock=bf3ea2428fa7114068c661106980b70ff4c8fd70 |
| message |  | string | @mock="" |

### 3.1.4 返回json串 示例

{

 "code": "",

 "data":

 {

 "siteId":"6b8b4690e7a727e3a330a308fe136a2e",

 "facePic": "http://seatent.com/1.jpg",

 "lvId": 2,

 "lvName": "Vip2",

 "memberId": bf3ea2428fa73434068c661106980b70ff4c8fd70,

"accountId": "bf3ea2428fa73434068c661106980b70ff4c8fd70"

 "mobile": "15858121765",

 "userName": "15858121765",

 "token": "bf3ea2428fa7114068c661106980b70ff4c8fd70"

 }

 ,

 "message": "",

 "result": 1,

 "success": true

}

## 3.2 商品类目

### 3.2.1 应用场景

获取商品列表前，可通过此接口先拿到分类，再以分类去搜索商品。

### 3.2.2 注意事项

按需调用，多次调用请间隔5秒以上。

### 3.2.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/goods/getTopCategory |
| 功能描述 | 查询海带平台中的所有顶级分类目录，按catId查询下级类目。 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| catId | 上级分类ID（0为顶级分类） | string | 必填，@mock= bf3ea2428fa73434068c661106980b70ff4c8fd70 |
| listShow | 显示分类（0显示全部，1允许显示的分类） | number | 选填，默认1 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| data |  | object |  |
| result|10 | --- | array<object> |  |
| image | ------分类图片 | string | @mock=$order('','fs:/attachment/goodscat/201511102028537343.png','fs:/attachment/goodscat/201511202143299202.png','fs:/attachment/goodscat/201511202143561167.png','fs:/attachment/goodscat/201511202144113701.png','fs:/attachment/goodscat/201601151139161696.png','fs:/attachment/goodscat/201511202144437000.png','fs:/attachment/goodscat/201511202144566199.png','','') |
| name | ------分类名称 | string | @mock=$order('奶粉','尿裤湿巾','营养辅食','喂养用品','宝宝护理','保健品','进口食品','成人日化','汽车用品','儿童服装') |
| parentId | ------分类上级id | string | @mock=$order(0,0,0,0,0,0,0,0,0,0) |
| id | ------分类id | string | @mock=$order(1300,1200,1100,0,0,0,0,0,0,0,0) |
| catOrder | ------分类排序 | number | @mock=$order(1100,1200,1300,1400,1500,1700,1800,1900,2000,0) |
| totalCount | ---总条数 | number | @mock=10 |
| code |  | string | @mock= |
| message |  | string | @mock= |
| result |  | number | @mock=1 |

### 3.2.4 返回json串 示例

{

 "code": "",

 "data":

 {

 "result": [

 {

 "catOrder": 1100,

 "id": 1300,

 "image": "",

 "name": "奶粉",

 "parentId": 0

 }

 ,

 {

 "catOrder": 0,

 "id": 0,

 "image": "",

 "name": "儿童服装",

 "parentId": 0

 }

 ],

 "totalCount": 10

 }

 ,

 "message": "",

 "result": 1

}

## 3.3 商品列表

### 3.3.1 应用场景

按贸易类型、分类、或商品名称搜索商品列表（一个商品一条记录）。

### 3.3.2 注意事项

1.接口调用需要间隔5秒一次。

### 3.3.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/goods/searchingGoods  |
| 功能描述 | 按分类获取商品列表（一个商品一条记录） |

说明：

1、sort字段、direct字段说明：

sort为空，按标签类型降序desc排序；

sort不为空，direct也必须不能为空，按direct字段sort排序；

2、tradeType

 1101：保税

 2202：一般贸易

 1303：直邮

 1404：海外自提

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| keyword | 关键字(商品名称或品牌或货号等） | string | 选填 |
| memberId | 用户ID | string | 必填@mock= bf3ea2428fa73434068c661106980b70ff4c8fd70 |
| appkey |  | number | 必填 |
| catName | 分类名称 | string | 选填 |
| catId | 分类编号 | string | 选填 |
| isFreepost | 是否包邮（1:是;0:否） | number | 选填@mock(1|0) |
| direct | 排序(升序asc，降序desc) | string | 选填 |
| sort | 排序字段（price：价格,buyNum：销量） | string | 选填 |
| pageSize | 页面大小数量 | number | 选填@mock=10，禁止超过50 |
| pageNum | 页面数量（第n页） | number | 选填@mock=1 |
| highPrice | 价格上限 | string | 选填 |
| brandName | 品牌名称 | string | 选填 |
| brandId | 品牌编号 | string | 选填 |
| tradeType | 贸易类型（1101，2202，3303，1404） | string | 选填,支持多个贸易类型以逗号隔开 |
| countryId | 产地编号 | string | 选填 |
| countryName | 国家名称 | string | 选填 |
| lowPrice | 价格下限 | number | 选填 |
| tradeTypeName | 贸易类型名称 | string | 选填 |
| ignoreStock | 是否忽略库存为0的商品 | string | 选填，传1显示库存为0商品 |
| needFacet | 是否需要搜索聚合信息（非必填） | number | 选填@mock=0 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| message |  | string | @mock="" |
| result | 结果(1:成功;0:失败) | number | @mock=1 |
| data |  | object |  |
| result|7 | 结果集 | array<object> |  |
| big | 商品大图 | string | @mock=http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg |
| brandId | 品牌编号 | string | @mock=$order(265,0,106,0,256,105,0) |
| brandName | 品牌名称 | string | @mock=$order('美林','undefined','迪尔玛','爱他美1','undefined') |
| buyNum | 销量（已售数量） | number | @mock=222 |
| catId | 分类编号 | string | @mock=$order(1101,1502,1110,1401,1802,1101,2029) |
| catName | 分类名称 | string | @mock=$order('爱他美','宝宝护肤','其他品牌','奶瓶奶嘴','饮品','爱他美','坐垫') |
| countryId | 产地编号 | number | @mock=$order(151,1,151,228,200,83,) |
| countryName | 产地名称 | string | @mock=$order('荷兰','其他','荷兰','美国','斯里兰卡','德国') |
| enableStore | 总库存 | number | @mock=$order(100,1000,93,37,211,109,940) |
| goodsId | 商品编号 | string | @mock=650 |
| hotSort | 商品热度 | number | 热门商品排序 0：非热门商品，>0 数值越大排名越靠前 |
| isFreepost | 是否包邮（1：是；0：否） | number | @mock=1 |
| life | 有效期（最近） | string | @mock=2018-06-30 |
| mktprice | 市场价格 | number | @mock=$order(282.9,338,50,650,1) |
| name | 商品名称 | string | @mock=“荷兰奶粉” |
| orderTagId | 订单标签编号 | number | @mock=1007 |
| originalPrice | 商品原价 | number | @mock=$order(5,11.1,12,110,32,557,15) |
| pCatId | 父分类编号 | string | @mock=$order(1100,1500,1100,1400,1800,1100,2028) |
| pCatName | 父分类名称 | string | @mock=奶粉 |
| SpecList|2 | 规格列表 | array<object> |  |
| num | ---规格数量 | number | @mock=2 |
| price | ---规格价格（有活动包含活动优惠） | number | @mock=-2 |
| price | 商品优惠后的最低金额 | number | @mock=100 |
| small | 小图片 | string | @mock=$order('http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg' |
| sn | 商品货号 | string | @mock=$order('HD1101002807','HD00002549','HD1110000005','HD1401000004') |
| tagBigLogo | 活动大logo | string | @mock=$order('http://img.seatent.com/statics/attachment/tagImage/201706191437428701.png' |
| tagId | 活动编号 | number | @mock=$order(264,1012,0,0,0,0,0) |
| tagLogo | 活动logo | string | @mock=$order('http://img.seatent.com/statics/attachment/tagImage/201706191437428701.png' |
| tagName | 活动名称 | string | @mock=”新品特惠” |
| tagSort | 活动排序 | number | @mock=$order(111,1,,,,,) |
| tagValid | 活动是否生效 | boolean | @mock=true |
| tax | 税率 | number | @mock=$order(11.9,0,10,10,0,0,0) |
| thumbnail | 缩略图 | string | @mock=$order('http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg |
| tiny | 微小图 | string | @mock=$order('http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg |
| tradeType | 贸易类型(1101保税、2202一般贸易、1303直邮、1404海外自提) | number | @mock=$order(2202, 1101, 1101, 1101, 2202, 1303, 2202) |
| tradeTypeName | 贸易类型名称 | string | @mock=$order('一般贸易','保税','保税','保税','一般贸易','直邮','一般贸易') |
| totalCount | ---总数量 | number | @mock=7 |
| pageSize | ---每页数量 | number | @mock=10 |
| facet | --- | object |  |
| tagIdList | ------标签编号列表 | array<string> | @mock=264 |
| tagNameList | ------标签名称列表 | array<string> | @mock='新品特惠' |
| tradeTypeNameList | ------贸易类型名称列表 | array<string> | @mock='一般贸易' |
| tradeTypeList | ------贸易类型列表 | array<string> | @mock='1' |
| countryIdList | ------产地编号列表 | array<string> | @mock='151' |
| countryNameList | ------产地名称列表 | array<string> | @mock='荷兰' |
| pcatIdList | ------父分类编号列表 | array<string> | @mock='1100' |
| pcatNameList | ------父分类名称列表 | array<string> | @mock='奶粉' |
| brandNameList | ------品牌名称列表 | array<string> | @mock='爱他美' |
| brandIdList | ------品牌编号列表 | array<string> | @mock='105' |
| brandCountList | ------品牌数量列表 | array<number> | @mock=1 |
| catIdList | ------分类编号列表 | array<string> | @mock='1101' |
| catNameList | ------分类名称列表 | array<string> | @mock='爱他美' |
| totalPageCount | ---总页面数 | number | @mock=1 |
| currentPageNo | ---当前页面no | number | @mock=1 |
| success |  | boolean | @mock=true |
| code | 错误码 | string | @mock="" |
|  |  |  |  |

### 3.3.4 返回json串 示例

{

 "code": "",

 "data":

 {

 "currentPageNo": 1,

 "facet":

 {

 "brandCountList": [

 1,

 1,

 1,

 1

 ],

 "brandIdList": [

 "105",

 "106",

 "256",

 "265"

 ],

 },

 "pageSize": 10,

 "result": [

 {

 "activeEnableStore": 10,

 "big": "测试内容j7us",

 "brandId": 0,

 "brandName": "undefined",

 "buyNum": 222,

 "catId": 1401,

 "catName": "奶瓶奶嘴",

 "countryId": 228,

 "countryName": "美国",

 "enableStore": 37,

 "goodsId": 256,

 "hotSort": 20584,

 "isFreepost": 0,

 "life": "测试内容pnk8",

 "mktprice": 338,

 "name": "thinkbaby不锈钢餐具套装（橙色）",

 "orderTagId": 0,

 "originalPrice": 110,

 "pCatId": 1400,

 "pCatName": "喂养用品",

 "preferentialPrice": 22865,

 "price": 85,

 "small": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg",

 "sn": "HD1401000004",

 "specList": [

 {

 "num": 2,

 "price": -2

 }

 ,

 "tagBigLogo": "undefined",

 "tagId": 0,

 "tagLogo": "undefined",

 "tagName": "undefined",

 "tagValid": true,

 "tax": 10,

 "thumbnail": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg",

 "tiny": "fs:/attachment/goods/201508201237529432\_tiny.jpg",

 "tradeType": 1101,

 "tradeTypeName": "保税"

 }

 ],

 "totalCount": 7,

 "totalPageCount": 1

 }

 ,

 "message": "",

 "result": 1,

 "success": true

}

## 3.4 获取商品详情

### 3.4.1 应用场景

根据传入的商品id获取商品详情（一个商品可能对应多个规格），以及每种规格的价格、库存、重量（包邮商品重量为0）、配货说明等详细信息。

### 3.4.2 注意事项

1.按需调用，调用间隔时间需5秒以上。

### 3.4.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/goods/getGoodsInfo |
| 功能描述 | 根据传入的商品id(商品sn)获取商品详情列表（一个商品可能对应多个规格） |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户编号 | string | 必填@mock=90908 |
| appkey |  | number | 必填@mock=21693225 |
| goodsId | 商品编号(2选1) | string | 必填@mock=11 |
| needCats | 是否需要类目(可选) | number | 选填@mock=0 |
| needIntro | 是否需要详情(可选) | number | 选填@mock=0 |
| needFocusImages | 是否需要焦点图(可选) | number | 选填@mock=1 |
| goodsSn | 商品货号(2选1，优先) | string | @mock=‘G323098’ |
| needSEO | 是否需要SEO(可选) | number | 选填@mock=1 |
| needImages | 是否需要橱窗图(可选) | number | 选填@mock=0 |
| onlyEnableStore | 仅需可用库存(可选) | number | 选填@mock=0 |
| needActivity | 是否需要促销信息（可选） | number | 选填@mock=0 |
| needParams | 是否需要商品参数(可选) | number | 选填@mock=1 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code | 错误码 | string | @mock='' |
| success |  | boolean | @mock=true |
| message |  | string | @mock='1' |
| data |  | object |  |
| Activities|2 | 促销 | array<Object> | 促销信息 |
| bigLogo | 限时折扣大图标，原tagBigLogo | string |  |
| endTime | 活动结束时间，原tagEndTime字段取消 | number | @mock=$order(1519833600000,1519833600000) |
| logo | 限时折扣小图标，原taglogo | string |  |
| name | 活动名称,原tagName字段取消 | string | @mock=$order('新品抢鲜','十月大促') |
| shortName | 活动段名称，原shortShortName字段 | string | @mock=特 |
| startTime | 活动开始时间,原tagBeginTime字段取消tagBeginTime | number | @mock=$order(1519833600000,1519833600000) |
| tagId | 活动Id | number | @mock=$order(15,100) |
| type | 活动类型 | number | 5-限时折扣 6-满减送@mock=$order(5,6) |
| lockEnable | ---是否可锁货 | number | @mock=1 |
| catName | ---类目名称 | string | @mock=爱他美 |
| metaKeywords | ---SEO关键字 | string |  |
| goodsId | ---商品编号 | string | @mock=11 |
| isFreePost | ---是否包邮(1:是;0:否) | number | @mock=0 |
| tagValid | ---标签是否生效 | boolean | @mock=true |
| metaDescription | ---SEO描述 | string |  |
| brandId | ---品牌编号 | string | @mock=105 |
| orderTagId | ---订单标签编号 | string | @mock=0 |
| barCode | ---商品条形码（多个以逗号分割） | string | @mock=31020387234 |
| seoKeywords | ---SEO平台搜索关键字 | string |  |
| params | ---参数 | string |  |
| sn | ---商品货号 | string | @mock=‘G1101000038’ |
| tagLogo | ---标签小图 | string |  |
| createTime | ---创建时间 | number | @mock=1519833600000 |
| tagName | ---促销标签名称 | string | @mock=限时特卖 |
| lastModifyDate | ---最后更新时间 | number | @mock=1430295823 |
| isBuyLimit | 是否达到购买上线0=不是，1=是 | number | @mock=1 |
| originalPurchase | 是否原价购买（活动原价购买） | boolean | 默认为false |
| images|9 | ---橱窗图列表（是否需要橱窗图） | array<object> |  |
| tinyPic | ------微小图 | string | @mock='http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg’ |
| thumbnail | ------缩略图 | string | @mock='http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg' |
| smallPic | ------小图 | string | @mock='http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg' |
| bigPic | ------大图地址 | string |  |
| isDefault | ------是否默认(1:是;0:否) | number |  |
| remark | ---备注 | string |  |
| tradeTypeName | ---贸易类型名称 | string | @mock=保税 |
| subTitle | ---副标题 | string | @mock=保税直供 |
| intro | ---商品详情 | string | @mock=’<p></p>’ |
| bigPic | ---大图 | string |  |
| tagBeginTime | ---促销标签开始时间 | number | @mock=1513242000000 |
| unit | ---数量单位 | string | @mock=罐 |
| countryId | ---国家编号 | string | @mock=83 |
| enableStore | ---商品总库存（只需要库存信息） | number | @mock=123 |
| weight | ---单件净重(克) | number | @mock=1100 |
| maxGroup | ---最大组数限制 | number | @mock=0 |
| disabled | ---是否禁用(0:不禁用;1:禁用) | number | @mock=0 |
| isSupply | ---是否在售(1:是;0:否) | number | @mock=0 |
| name | ---商品名称 | string | @mock=德国爱他美白金版pre段800g |
| pageTitle | ---SEO标题 | string | @mock=爱他美奶粉 |
| min | ---最小件数限制 | number | @mock=0 |
| tagEndTime | ---促销标签结束时间 | number | @mock=1513242000000 |
| tinyPic | ---微小图片 | string |  |
| minGroup | ---最小组数限制 | number | @mock=0 |
| thumbnail | ---缩略图 | string | @mock='http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg' |
| cats | ---类目层级列表（是否需要类目） | array<object> |  |
| catOrder | ------类目排序 | number |  |
| parentId | ------父类目编号 | string |  |
| name | ------类目名称 | string |  |
| catPath | ------类目编号路径 | string |  |
| catId | ------类目编号 | string |  |
| expirationDate | ---有效期限(最近日期) | number | @mock=1519833600000 |
| tradeType | ---贸易类型(1101保税、2202一般贸易、1303直邮、1404海外自提) | number | @mock=1101 |
| smallPic | ---小图 | string |  |
| brandName | ---品牌名称 | string | @mock=爱他美 |
| orderDaily | ---单人单日订单限制 | number | @mock=0 |
| tagShortName | ---标签段名称 | string | @mock=特 |
| countryName | ---国家名称 | string | @mock=德国 |
| tagBigLogo | ---标签大图 | string |  |
| updateTime | ---更新时间 | number | @mock=1481523548000 |
| catId | ---类目编号 | string | @mock=1101 |
| tagId | ---促销标签编号 | number | @mock=284 |
| tax | ---税率 | number | @mock=10 |
| max | ---最大个数限制 | number | @mock=0 |
| shelfLife | ---商品保质期 | number | @mock=12 |
| result | 结果（1:成功;0:失败） | number | @mock=1 |

### 3.4.3 返回json串 示例

{

 "code": "''",

 "data":

 {

 "activities": [

 {

 "bigLogo": "测试内容f3nr",

 "endTime": 1519833600000,

 "logo": "测试内容l968",

 "name": "新品抢鲜",

 "shortName": "特",

 "startTime": 1519833600000,

 "tagId": 15,

 "type": 5

 }

 ],

 "barCode": "31020387234",

 "bigPic": "测试内容5l1d",

 "brandId": 105,

 "brandName": "爱他美1",

 "canCreate": true,

 "catId": 1101,

 "catName": "爱他美",

 "cats": [

 {

 "catId": 88874,

 "catOrder": 83306,

 "catPath": "测试内容5812",

 "name": "测试内容7pb7",

 "parentId": 14063

 }

 ],

 "countryId": 83,

 "countryName": "德国",

 "createTime": 1519833600000,

 "disabled": 0,

 "enableStore": 123,

 "expirationDate": 1519833600000,

 "goodsId": 11,

 "images": [

 {

 "bigPic": "测试内容762o",

 "isDefault": 41726,

 "smallPic": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg",

 "thumbnail": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg",

 "tinyPic": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg"

 }

 ,

 {

 "bigPic": "测试内容762o",

 "isDefault": 41726,

 "smallPic": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg",

 "thumbnail": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg",

 "tinyPic": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg"

 }

 ],

 "intro": "<p><img align=\"absmiddle\" src=\"http://gd1.alicdn.com/imgextra/i1/2482209074/TB2m6BGcpXXXXcEXpXXXXXXXXXX\_!!2482209074.jpg\" style=\"border: 0px; vertical-align: top; margin: 0px; padding: 0px; color: rgb(0, 0, 0); font-family: tahoma, arial, 宋体, sans-serif; font-size: 14px; line-height: 21px;\"/><img align=\"absmiddle\" src=\"http://gd3.alicdn.com/imgextra/i3/2482209074/TB2xNoxcpXXXXXfXXXXXXXXXXXX\_!!2482209074.jpg\" style=\"border: 0px; vertical-align: top; margin: 0px; padding: 0px; color: rgb(0, 0, 0); font-family: tahoma, arial, 宋体, sans-serif; font-size: 14px; line-height: 21px; float: none;\"/></p>",

 "isBuyLimit": 0,

 "isFreePost": 0,

 "isSupply": 0,

 "lastModifyDate": 1430295823,

 "lockEnable": 1,

 "marketEnable": 1,

 "max": 0,

 "maxGroup": 0,

 "metaDescription": "测试内容a43i",

 "metaKeywords": "测试内容3muh",

 "min": 0,

 "minGroup": 0,

 "mktPrice": 389,

 "name": "德国爱他美白金版pre段800g",

 "orderDaily": 0,

 "orderTagId": 0,

 "originalPurchase": false,

 "pageTitle": "爱他美奶粉",

 "params": "测试内容76t9",

 "remark": "测试内容g7qn",

 "seoKeywords": "测试内容3p98",

 "shelfLife": 12,

 "smallPic": "测试内容w1v7",

 "sn": "‘G1101000038’",

 "subTitle": "测试内容o454",

 "tax": 10,

 "thumbnail": "http://img.seatent.com/statics/attachment/goods/201605131732556681\_small.jpg",

 "tinyPic": "测试内容o77v",

 "tradeType": 1101,

 "tradeTypeName": "保税",

 "unit": "罐",

 "updateTime": 1481523548000,

 "weight": 1100

 }

 ,

 "message": "'1'",

 "result": 1,

 "success": true

}

## 3.5 获取商品动态价格

### 3.5.1 应用场景

根据传入条件获取商品动态价格。（进入商品详情页时获取一次动态价格，选择其他条件同样需要请求动态价格。）

### 3.5.2 注意事项

1. 请按需调用；
2. 动态价格会根据传入的参数变化；
3. 当minPrice = maxPrice 时价格为唯一确定值。（可用于下单）；
4. 如果不传cityId：1.默认该用户保存的默认地址 2.最后一次订单的地址 3.杭州市的cityId；
5. enableStore=0,表示该组合缺货；

### 3.5.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/goods/getGoodsPrice |
| 功能描述 | 根据传入的条件获取商品动态价格（地区，生产日期，购买数量，规格等） |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| appkey |  | string | 必填 |
| memberId | 用户ID | string | 必填； |
| goodsId | 商品ID | string | 必填；与goodsSn二选一@mock=1234 |
| goodsSn | 商品编号 | string | 必填；与goodsId二选一@mock=“HD1212” |
| num | 购买数量 | number | 必填；@mock=2 |
| productNum | 规格数量 | number | 选填；@mock=2 |
| cityId | 配送城市ID | number | 选填；@mock=310000 |
| life | 生产日期 | string | 选填；@mock=2017-01 |

返回结果：

|  |  |  |  |
| --- | --- | --- | --- |
| **变量名** | **含义** | **类型** | **备注** |
| code | 错误码 | string |  |
| message | 错误信息 | string |  |
| result | 是否成功 | number | 成功返回1，失败返回0 |
| success | 是否成功 | boolean | 成功返回true，失败返回false |
| data |  | object |  |
|  canCreate | 是否可以创建订单 | number | @mock=1 |
|  enableStore | 库存 | number | @mock=200; |
|  isActivity | 是否是活动信息，用于详情页判断是否是活动库存 | number | @mock=0, 0不是，1是 |
| tagPreferential | 限时折扣优惠金额，默认为0 | number | @mock=10 |
| isCard | 是否需要身份证照片 | number | 0是不需要，1是需要 |
| lifeType | 效期类型 | number | 1是生产日期，2是有效期 |
| lifes|6 | （保税的商品不需要取） | array<Object> |  |
| enable | 是否可用 | boolean | @mock=$order(true,true,false,true,false,false) |
| life | 有效期 | string | @mock=$order('2017-09','2017-11','2017-10','2017-12','2017-11','2017-10') |
| maxPrice | 单规格最高价 | number | @mock=300 |
| minPrice | 单规格最低价 | number | @mock=200 |
| specs|6 |  | array<object> |  |
| enable |  |  |  |
| isFreePost | 是否包邮（暂时不用） | number | 0是不包邮，1是包邮（实际运费参照运费接口） |
| productName | 规格名称 | string | @mock=$order('1罐装','2罐装','3罐装','4罐装') |
| productNum | 规格数量 | number | mock=$order(1,2,3,4,5,6) |
| unit | 单位 | string | @mock=$order('罐','罐','罐','罐','罐','罐') |

### 3.5.4 返回json串 示例

{

 "code": "测试内容ykwa",

 "data":

 {

 "canCreate": 1,

 "enableStore": 208,

 "isActivity": false,

 "lifes": [

 {

 "enable": true,

 "life": "2017-09"

 }

 ,

 {

 "enable": true,

 "life": "2017-11"

 }

 ],

 "maxPrice": 300,

 "minPrice": 200,

 "specs": [

 {

 "enable": true,

 "isFreePost": 0,

 "productName": "1罐装",

 "productNum": 1,

 "unit": "罐"

 }

 ,

 {

 "enable": true,

 "isFreePost": 0,

 "productName": "2罐装",

 "productNum": 2,

 "unit": "罐"

 }

 ,

 {

 "enable": true,

 "isFreePost": 0,

 "productName": "3罐装",

 "productNum": 3,

 "unit": "罐"

 }

 ],

 "tagPreferential": 10

 }

 ,

 "message": "测试内容5016",

 "result": 1,

 "success": true

}

## 3.6 商品多规格信息

### 3.6.1 应用场景

获取商品多规格基础信息。

### 3.6.2 注意事项

1. 此接口只用于显示规格。
2. 请按需调用。

### 3.6.3API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | api/v2/goods/getGoodsSpecs |
| 功能描述 | 获取商品多规格基础信息。 |

输入参数：

| 变量名 | 含义 | 类型 | 备注 |
| --- | --- | --- | --- |
| memberId | 用户id | string | 必填，@mock=985 |
| goodsId | 商品Id | string | 必填，@mock=8270 |
| goodsSn | 商品货号(商品Id 2选1) | string | 必填，@mock=HD1802003213 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| success |  | boolean | @mock=true |
| data|3 |  | array<object> |  |
| productName | ---规格名称 | string | @mock=$order('1瓶装','2瓶装','3瓶装') |
| average | ---规格平均单价 | number | @mock=$order(1.02,0.51,0.34) |
| tax | ---税率 | number | @mock=$order(0,0,0,0) |
| weight | ---毛重 | number | @mock=$order(1000,2000,3000,4000) |
| productNum | ---规格数量 | number | @mock=$order(1,2,3) |
| price | ---售价 | number | @mock=$order(1.02,1.02,1.02) |
| unit | ---单位 | string | @mock=$order('瓶','瓶','瓶') |
| isFreePost | ---是否包邮 | number | @mock=$order(0,0,0) |
| code |  | string |  |
| result |  | number | @mock=1 |
| message |  | string |  |

### 3.5.4 返回json串 示例

{

 "code": "测试内容7p4j",

 "data": [

 {

 "average": 1.02,

 "enableStore": 77,

 "isFreePost": 0,

 "life": "2017-02-01 00:00:00",

 "price": 1.02,

 "productName": "1瓶装",

 "productNum": 1,

 "tax": 0,

 "unit": "瓶",

 "weight": 1000

 }

 ,

 {

 "average": 0.51,

 "enableStore": 77,

 "isFreePost": 0,

 "life": "2017-02-01 00:00:00",

 "price": 1.02,

 "productName": "2瓶装",

 "productNum": 2,

 "tax": 0,

 "unit": "瓶",

 "weight": 2000

 }

 ],

 "message": "测试内容hg2s",

 "result": 1,

 "success": true

}

## 3.7 获取运费

### 3.7.1 应用场景

获取商品运费

### 3.7.2 注意事项

1. 按需调用，禁止轮询；

2. cityId需要传海带认可的地区Id，请以“获取地区信息接口”返回的数据为准；

3. 多个商品货号以逗号分隔的时候，需要对url进行encode utf-8处理；

### 3.7.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | api/v2/goods/getGoodsFreight |
| 功能描述 | 获取商品运费 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| goodsSns | 商品货号，逗号隔开 | string | 必填；goodsIds参数2选1 |
| goodsIds | 商品ID，逗号隔开 | string | 必填；goodsSns参数2选1 |
| memberId | 客户ID | string | 必填； |
| cityId | 城市ID | number | 必填； |
| productNums | 规格数量，逗号隔开 | number | 必填； |
| lifes | 生产日期，逗号隔开 | string | 必填 |
| nums | 数量，逗号隔开 | number | 必填； |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| data |  | object |  |
| fee | ---运费 | number | @mock=2 |
| success |  | boolean | @mock=true |
| code |  | string | @mock= |
| result |  | number | @mock=1 |
| message |  | string | @mock= |

### 3.7.2 返回json串 示例

{
    "code": "",
    "data":
    {
        "fee": 2
    }
    ,
    "message": "",
    "result": 1,
    "success": true
}

## 3.8 创建订单

### 3.8.1 应用场景

创建订单

### 3.8.2 注意事项

1. 由于参数比较多因此出现问题请先检查入参字段名拼写是否正确。
2. 看清2选1的参数或互斥参数，传参时仅可传一个。
3. 当地址出现中文或参数值中有标点符号的时候需要url encode -utf8转码，否则可能出现授权认证失败等错误。
4. 创建订单成功会返回1条或多条订单信息
5. 当返回多条订单信息时是由于：贸易类型不同、商品供应商不同、跨境贸易单笔订单金额超5000 等原因拆单造成。

### 3.8.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/order/createOrders |
| 功能描述 | 创建订单 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| area | 区域编码 | string | 必填 ;从地区表中获取的reginId(第三级区级) @mock=310000 |
| goodsSns | 商品货号SNs | string | 必填，与goodsIds二选一，以逗号隔开 |
| nums | 数量 | string | 必填;以逗号隔开 |
| name | 用户名称 | string | 必填 ; |
| productNums | 规格数量 | string | 必填;以逗号隔开 @mock=1 |
| mobile | 手机 | string | 必填  |
| appkey | 应用平台appkey | string | 必填 @mock=63568028 |
| address | 地址 | string | 必填 详细地址;  |
| memberId | 会员ID | string | 必填; |
| goodsIds | 商品ID | string | 必填 与goodsSns二选一，以逗号隔开; @mock=8309 |
| identification | 身份证号码 | string | 必填 @mock=362502199106073246 |
| thirdBuyAmount | 代购价格 | number | 保税商品必填，非保税商品不需要填写 @mock=89.9 |
| thirdOrderNo | 保税商品订单号 | string | 对接保税收款的第三方必填，其他的不需要填写(2020.1.8) |
| lifes | 生产日期 | string | 选填；保税商品不需要，以逗号隔开;@mock=2018-04 |
| paymentId | 支付Id | number | 选填 ; |
| customOrder | 第三方订单号 | string | 选填 |
| remarks | 买家备注 | string | 选填; #;#隔开 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| result |  | number | @mock=1 |
| data|2 |  | object |  |
| orderAmount | ------订单金额 | number | @mock=135 |
| taxAmount | ------税额 | number | @mock=0 |
| sn | ------订单编号 | string | @mock=20170707000019 |
| shippingAmount | ------运费 | number | @mock=0 |
| needPayMoney | ------需要支付金额 | number | @mock=135 |
| memberId | ------客户ID | string | @mock=985 |
| orderId | ------订单ID | string | @mock=533913 |
| goodsNum | ------商品数量 | number | @mock=1 |
| orderSource | ------订单来源 | number | @mock=0 |
| paymentId | ------支付方式ID | string | @mock=0 |
| preferential | ------优惠金额 | number | @mock=0 |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| message |  | string | @mock= |

### 3.8.4 返回json串 示例

{

"code": "",

 "data": {

 "orderList": [

 {

 "goodsNum": 1,

 "memberId": "985",

 "needPayMoney": 135,

 "orderAmount": 135,

 "orderId": "533914",

 "orderSource": 6,

 "paymentId": "0",

 "preferential": 0,

 "shippingAmount": 0,

 "sn": "201707070000190",

 "taxAmount": 11

 },

 {

 "goodsNum": 1,

 "memberId": "985",

 "needPayMoney": 135,

 "orderAmount": 135,

 "orderId": "533914",

 "orderSource": 6,

 "paymentId": "0",

 "preferential": 0,

 "shippingAmount": 0,

 "sn": "201707070000190",

 "taxAmount": 11

 }

 ]

 },

 "message": "",

 "result": 1,

 "success": true

}

## 3.9 获取订单列表

### 3.9.1 应用场景

根据传入的订单id获取订单详情列表。

### 3.9.2 注意事项

1. 调用接口需间隔30分钟调用一次。

### 3.9.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/order/orderList |
| 功能描述 | 根据传入的订单id列表（以逗号分割）获取订单详情列表 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| pageSize | 每页条数 | number | 必填; |
| memberId | 会员ID | string | 必填; |
| pageNum | 第几页 | number | 必填; |
| mobile | 收货电话 | string | 选填 |
| abnormal | 异常订单 | number | 选填;0，正常订单；1，异常订单(异常)；2，异常处理中 |
| goodsSn | 商品货号 | string | 选填 |
| hasShipInfo | 是否有物流信息 | number | 选填; 0-没有 1-有 |
| endTime | 结束时间 | string | 选填; |
| orderSn | 订单号 | string | 选填; |
| customOrder | 第三方订单号 | string | 选填; |
| status | 要查询的订单 | number | 选填 全部 （不传），待付款-0，待发-2，待收货-5 完成-7; 已取消-8； |
| orderType | 订单类型 | number | 选填 0-全部 1-发货订单 |
| timeType | 时间类型 | number | 选填 1-下单时间、2-支付时间、3-发货时间、4-收货时间 |
| tradeType | 贸易类型 | number | 选填; 1101保税、2202一般贸易、1303直邮、1404海外自提 |
| beginTime | 开始时间 | string | 选填; |
| receiver | 收货人姓名 | string | 选填; |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code | 错误码 | string | @mock= |
| result |  | number | @mock=1 |
| message |  | string | @mock="" |
| data |  | object |  |
| totalPageCount | ---总页码数 | number | @mock=1 |
| result|8 | --- | array<object> |  |
| remark | ------备注 | string | @mock= |
| abnormalStatus | ------异常状态 | number | @mock=0 |
| applyReason | ------申请售后理由 | string | @mock=商品破损 |
| orderStatus | ------订单状态 | string | @mock=待付款 |
| id | ------订单ID | string | @mock=532932 |
| orderAmount | ------订单金额 | number | @mock=12 |
| orderSource | ------订单来源；锁货订单=11 | number | @mock=(1,4,5,11,2) |
| cancelReason | ------取消理由 | string | @mock=拍重复了 |
| payTradeNo | ------支付单号 | string | @mock=458569670 |
| tradeTypeName | ------贸易类型名称 | string | @mock=跨境贸易 |
| shipName | ------收货人姓名 | string | @mock=“测试” |
| needPayTax | ------需要支付税额 | number | @mock=0 |
| createTime | ------创建时间 | number | @mock=1492999429 |
| commission | ------佣金 | number | @mock=0 |
| needPayMoney | ------需要支付金额 | number | @mock=12 |
| paymoney | ------支付金额 | number | @mock=0 |
| shipMobile | ------收货人电话 | string | @mock=18757106157 |
| abnormalType | ------异常类型 | number | @mock=0 |
| customOrder | ------第三方订单号 | string | @mock=3474865960 |
| paySuccessTime | ------支付成功时间 | number | @mock=0 |
| paymentName | ------支付方式名称 | string | @mock=支付宝 |
| hasShipInfo | ------是否有物流信息 | number | @mock=0 |
| memberId | ------会员ID | string | @mock=3812 |
| shipStatus | ------发货状态 | number | @mock=2 |
| abnormal | ------是否为异常订单0，正常；1,异常；2,售后； | number | @mock=0 |
| orderItemList|1-3 | ------ | array<object> |  |
| tagId | ---------活动ID | number | @mock=0 |
| num | ---------数量 | number | @mock=1 |
| preferential | ---------优惠金额 | number | @mock=0 |
| name | ---------商品名称 | string | @mock=荷兰牛栏5段800g |
| tax | ---------税率 | number | @mock=11.9 |
| unit | ---------单位 | string | @mock=罐 |
| lockSn | ---------锁货单号 | string | @mock= |
| weight | ---------重量 | number | @mock=0 |
| sn | ---------商品货号 | string | @mock=321564564 |
| productNum | ---------规格数量 | number | @mock=1 |
| lifeStr | ---------保质期 | string | @mock=2017-05 |
| price | ---------价格 | number | @mock=12 |
| image | ---------商品图片 | string | @mock=http://dev.seatent.com/statics/attachment/goods/201505041019304784\_thumbnail.jpg |
| productName | ---------规格名称 | string | @mock=1件装 |
| goodsId | ---------商品ID | string | @mock=18 |
| originPrice | 商品原价 | Number | @mock=77 |
| shippingAmount | ------邮费 | number | @mock=0 |
| ships | ----物流信息 | array<object> |  |
| dlyCode | ------快递代码 | string | @mock=ems |
| shipNo | ------快递单号 | string | @mock=1100011355006|9769882836001 |
| sn | ------订单编号 | string | @mock=201704249251460 |
| depotName | ------仓库名称 | string | @mock=诚淘测试仓库 |
| preferential | ------优惠金额 | number | @mock=0 |
| goodsAmount | ------商品金额 | number | @mock=12 |
| weight | ------重量 | number | @mock=0 |
| goodsNum | ------商品数量 | number | @mock=1 |
| addressId | ------客户地址ID | string | @mock=94789 |
| idNumber | ------身份证号 | string | @mock=330782198908230611 |
| abnormalSource | ------异常来源1，用户；2，管理员；3，供应商； | number | @mock=1 |
| payStatus | ------支付状态 | number | @mock=0 |
| status | ------状态 | number | @mock=0 |
| tradeType | ------贸易类型 | number | @mock=1101 |
| taxAmount | ------税额 | number | @mock=1 |
| saleCmplTime | ------发货时间 | number | @mock=0 |
| currentPageNo | ---当前页码 | number | @mock=1 |
| totalCount | ---总记录量 | number | @mock=1 |
| pageSize | ---页面大小 | number | @mock=10 |

### 3.9.4 返回json串 示例

{
    "code": "",
    "data":
    {
        "currentPageNo": 1,
        "pageSize": 10,
        "result": [
        {
            "abnormal": 0,
            "abnormalSource": 1,
            "abnormalStatus": 0,
            "abnormalType": 0,
            "addressId": 94789,
            "applyReason": "商品破损",
            "cancelReason": "拍重复了",
            "commission": 0,
            "createTime": 1492999429,
            "customOrder": "3474865960",
            "depotName": "诚淘测试仓库",
            "dlyCode": "ems",
            "dlyName": "EMS邮政快递",
            "goodsAmount": 12,
            "goodsNum": 1,
            "hasShipInfo": 0,
            "id": 532932,
            "idNumber": "330782198908230611",
            "memberId": 3812,
            "needPayMoney": 12,
            "needPayTax": 0,
            "orderAmount": 12,
            "orderItemList": [
            {
                "goodsId": 18,
                "image": "http://dev.seatent.com/statics/attachment/goods/201505041019304784\_thumbnail.jpg",
                "lifeStr": "2017-05",
                "lockSn": "",
                "name": "荷兰牛栏5段800g",
                "num": 1,
                "orderTagId": 0,
                "originPrice": 75,
                "preferential": 0,
                "price": 12,
                "productName": "1件装",
                "productNum": 1,
                "sn": "321564564",
                "tagId": 0,
                "tax": 11.9,
                "unit": "罐",
                "weight": 0
            }
            ],
            "orderSource": 2,
            "orderStatus": "待付款",
            "payStatus": 0,
            "paySuccessTime": 0,
            "payTradeNo": "458569670",
            "paymentName": "支付宝",
            "paymoney": 0,
            "preferential": 0,
            "remark": "",
            "saleCmplTime": 0,
            "shipMobile": "18757106157",
            "shipName": "“测试”",
            "shipNo": "1100011355006|9769882836001",
            "shipStatus": 2,
            "shippingAmount": 0,
            "sn": "201704249251460",
            "status": 0,
            "taxAmount": 1,
            "tradeType": 1101,
            "weight": 0
        }

],
        "totalCount": 1,
        "totalPageCount": 1
    }
    ,
    "message": "",
    "result": 1
}

## 4.0 获取订单详情

### 4.0.1 应用场景

根据传入的订单id或sn获取订单详情。

### 4.0.2 注意事项

1. 调用接口按需调用；
2. 订单所有状态：case -7 = "已换货";case -6= "换货被拒绝";case -5= "拒绝退换货";case -4="申请换货";case -3 = "已申请退换货";case -2 = "同意退换货";case -1= "退款";case 0 = "待付款";case 1 = "已付款待确认";case 2 = "已付款";case 3= "配货中";case 4= "未发货";case 5= "已发货";case 6= "已收货";case 7= "已完成";case 8= "已取消";case 9 = "订单已生效";case 10 = "拆单完成,待设置收货地址";case 11= "海关审核中";default= "错误状态";

### 4.0.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/order/orderDetail |
| 功能描述 | 根据传入的订单id订单详情 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | string | 必填 |
| customOrder | 第三方订单号 | string | 与订单SN，订单ID 3选1 |
| orderSn | 订单sn | string | 与订单ID，第三方订单号 3选1 |
| orderId | 订单ID | string | 与订单sn，第三方订单号 3选1 |
| needAddress | 是否需要返回地址信息 | number | 选填，默认不取，需要时参数=1 |
| needStoreInfo | 是否需要返回供应商信息 | number | 选填，默认不取，需要时参数=1 |
| needPayment | 是否需要返回支付信息 | number | 选填，默认不取，需要时参数=1 |
| needGoods | 是否需要返回商品信息 | number | 选填，默认不取，需要时参数=1 |
| needAbnormal | 是否需要异常状态 | number | 选填，默认不取，需要时参数=1 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| needIDCardPhoto | 是否需要完善信息 | number | 0 不用完善信息 1 需要完善信息 |
| message |  | string | @mock="" |
| needIDCard | 是否需要身份证号码 | boolean | @mock=true |
| data |  | object |  |
| idcardPhoto | ---身份证正面照片 | string | @mock=http://mall.test.seatent.com/statics/attachment/store/goods/201707171550316978.jpg |
| abnormalType | ---异常类型 -1=未处理 0-拒绝 1-退货 2-换货 3-补发 | number | @mock=0 |
| shipName | ---收件人姓名 | string | @mock=“测试” |
| declareStatus | ---报关状态 | number | @mock=0 |
| taxAmount | ---税费 | number | @mock=1 |
| memberId | ---用户ID | string | @mock=4185 |
| tradeType | ---贸易类型 | number | @mock=1101 |
| orderId | ---订单ID | string | @mock=532939 |
| abnormalStatus | ---异常状态 | number | @mock=0 |
| depotName | ---仓库名称 | string | @mock=诚淘测试仓库 |
| orderAmount | ---订单金额 | number | @mock=77 |
| preferential | ---优惠金额 | number | @mock=30 |
| goodsAmount | ---商品金额 | number | @mock=70 |
| needPayTax | ---需支付税额 | number | @mock=0 |
| paymentName | ---支付名称 | string | @mock=银行卡支付 |
| shipZip | ---收货人邮编 | string | @mock=310006 |
| shipStatus | ---发货状态 | number | @mock=2 |
| shipMobile | ---收件人手机号 | string | @mock=18757106157 |
| orderItemList|1-3 | ---订单明细 | array<object> |  |
| lifeStr | ------有效期 | string | @mock=2017-05 |
| unit | ------单位 | string | @mock=罐 |
| preferential | ------优惠金额 | number | @mock=0 |
| goodsId | ------商品ID | string | @mock=1110 |
| tax | ------税率 | number | @mock=0 |
| orderTagId | ------订单活动ID | number | @mock=0 |
| price | ------商品单价 | number | @mock=70 |
| weight | ------重量 | number | @mock=0 |
| productName | ------规格名称 | string | @mock=1件装 |
| image | ------商品图片 | string | @mock=http://dev.seatent.com/statics/attachment/goods/201505041019304784\_thumbnail.jpg |
| originPrice | ------商品原价 | number | @mock=75 |
| sn | ------商品货号 | string | @mock=321564564 |
| num | ------商品数量 | number | @mock=1 |
| name | ------商品名称 | string | @mock=荷兰牛栏5段800g |
| lockSn | ------锁货 单号 | string | @mock=“sh3285825” |
| tagId | ------商品标签ID | number | @mock=0 |
| productNum | ------规格数量 | number | @mock=1 |
| customOrder | ---小B客户导入订单号 | string | @mock=3474865960 |
| paymoney | ---支付金额 | number | @mock=77 |
| sn | ---订单编号 | string | @mock=20170424925336 |
| createTime | ---创建时间 | number | @mock=1483954583 |
| shippingAmount | ---运费 | number | @mock=10 |
| weight | ---重量 | number | @mock=0 |
| ships | 物流信息 | array<object> |  |
| dlyCode | ---快递公司代号 | string | @mock=ems |
| shipNo | ---运单号 | string | @mock=1100011355006|9769882836001 |
| abnormal | ---是否异常订单 | number | @mock=@int(0,1,2) |
| tradeTypeName | ---贸易类型名称 | string | @mock=跨境贸易 |
| completeTime | ---订单完成时间 | number | @mock=1483954583 |
| remark | ---备注 | string | @mock=合并付款:(532937,532936) |
| commission | ---佣金 | number | @mock=0 |
| orderSourceName | ---订单来源（字符串） | string | @mock=分销 |
| status | ---订单状态 | number | @mock=0 |
| applyReason | ---申请原因 | string | @mock=商品破损 |
| returnDes | ---申请售后描述 | string | @mock="商品破损" |
| allocationTime | ---接单时间 | number | @mock=0 |
| saleCmplTime | ---发货时间 | number | @mock=0 |
| cancelReason | ---取消原因 | string | @mock=拍重复了 |
| abnormalSource | ---异常来源 | number | @mock=0 |
| paySuccessTime | ---支付时间 | number | @mock=1493018561 |
| shipAddr | ---收货地址 | string | @mock=“测试地址” |
| idNumber | ---身份证号 | string | @mock=330782198908230611 |
| payStatus | ---支付状态 | number | @mock=2 |
| goodsNum | ---商品数量 | number | @mock=1 |
| orderStatus | ---订单状态 | string | @mock=待付款 |
| payTradeNo | ---交易号 | string | @mock=000044526020 |
| hasShipInfo | ---是否有物流信息 | number | @mock=0 |
| progress | ---进度条位置 | number | @mock=1 |
| needPayMoney | ---需支付金额 | number | @mock=77 |
| idcardBackPhoto | ---身份证背面照片 | string | @mock=http://mall.test.seatent.com/statics/attachment/store/goods/201707171550316978.jpg |
| orderSource | ---订单来源 | number | @mock=6 |
| addressId | ---客户地址ID | string | @mock=4583 |
| shippingArea | ---收货区域 | string | @mock=浙江省-杭州市-下城区 |
| code | 错误码 | string | @mock= |
| result |  | number | @mock=1 |
| success |  | boolean | @mock=true |

### 4.0.4 返回json串 示例

{
    "code": "",
    "data":
    {
        "abnormal": 1,
        "abnormalSource": 0,
        "abnormalStatus": 0,
        "abnormalType": 0,
        "addressId": 4583,
        "allocationTime": 0,
        "applyReason": "商品破损",
        "cancelReason": "拍重复了",
        "commission": 0,
        "completeTime": 1483954583,
        "createTime": 1483954583,
        "customOrder": "3474865960",
        "declareStatus": 0,
        "depotName": "诚淘测试仓库",
        "dlyCode": "ems",
        "dlyName": "EMS邮政快递",
        "goodsAmount": 70,
        "goodsNum": 1,
        "hasShipInfo": 0,
        "idNumber": "330782198908230611",
        "idcardBackPhoto": "http://mall.test.seatent.com/statics/attachment/store/goods/201707171550316978.jpg",
        "idcardPhoto": "http://mall.test.seatent.com/statics/attachment/store/goods/201707171550316978.jpg",
        "memberId": 4185,
        "needPayMoney": 77,
        "needPayTax": 0,
        "orderAmount": 77,
        "orderId": 532939,
        "orderItemList": [
        {
            "goodsId": 1110,
            "image": "http://dev.seatent.com/statics/attachment/goods/201505041019304784\_thumbnail.jpg",
            "lifeStr": "2017-05",
            "lockSn": "“sh3285825”",
            "name": "荷兰牛栏5段800g",
            "num": 1,
            "orderTagId": 0,
            "originPrice": 75,
            "preferential": 0,
            "price": 70,
            "productName": "1件装",
            "productNum": 1,
            "sn": "321564564",
            "tagId": 0,
            "tax": 0,
            "unit": "罐",
            "weight": 0
        }
        ,
        {
            "goodsId": 1110,
            "image": "http://dev.seatent.com/statics/attachment/goods/201505041019304784\_thumbnail.jpg",
            "lifeStr": "2017-05",
            "lockSn": "“sh3285825”",
            "name": "荷兰牛栏5段800g",
            "num": 1,
            "orderTagId": 0,
            "originPrice": 75,
            "preferential": 0,
            "price": 70,
            "productName": "1件装",
            "productNum": 1,
            "sn": "321564564",
            "tagId": 0,
            "tax": 0,
            "unit": "罐",
            "weight": 0
        }
        ],
        "orderSource": 6,
        "orderSourceName": "分销",
        "orderStatus": "待付款",
        "payStatus": 2,
        "paySuccessTime": 1493018561,
        "payTradeNo": "000044526020",
        "paymentName": "银行卡支付",
        "paymoney": 77,
        "preferential": 30,
        "progress": 1,
        "remark": "合并付款:(532937,532936)",
        "returnDes": "商品破损",
        "saleCmplTime": 0,
        "shipAddr": "“测试地址”",
        "shipMobile": "18757106157",
        "shipName": "“测试”",
        "shipNo": "1100011355006|9769882836001",
        "shipStatus": 2,
        "shipZip": "310006",
        "shippingAmount": 10,
        "shippingArea": "浙江省-杭州市-下城区",
        "sn": "20170424925336",
        "status": 0,
        "taxAmount": 1,
        "tradeType": 1101,
        "tradeTypeName": "跨境贸易",
        "weight": 0
    }
    ,
    "message": "",
    "needIDCard": true,
    "needIDCardPhoto": 83836,
    "result": 1,
    "success": true
}

## 4.1 获取地区信息

### 4.1.1 应用场景

获取海带对应级别的地区ID和名称信息

### 4.1.2 注意事项

1. 得到海带地区列表如发现有不一致的情况，可能原因：海带：XX区 三方平台：XX县；
2. 如果发现有缺失的地址请联系海带技术支持，技术支持会根据情况更新海带地区数据；

### 4.1.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | [http://img.pre.seatent.com](http://api.pre.seatent.com)/statics/json/regionsAll.json |

### 4.1.4 返回json串 示例

{

 "110000": {

 "areaId": 2,

 "children": {

 "110100": {

 "areaId": 2,

 "children": {

 "110101": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110101,

 "regionName": "东城区"

 },

 "110102": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110102,

 "regionName": "西城区"

 },

 "110105": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110105,

 "regionName": "朝阳区"

 },

 "110106": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110106,

 "regionName": "丰台区"

 },

 "110107": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110107,

 "regionName": "石景山区"

 },

 "110108": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110108,

 "regionName": "海淀区"

 },

 "110109": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110109,

 "regionName": "门头沟区"

 },

 "110111": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110111,

 "regionName": "房山区"

 },

 "110112": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110112,

 "regionName": "通州区"

 },

 "110113": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110113,

 "regionName": "顺义区"

 },

 "110114": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110114,

 "regionName": "昌平区"

 },

 "110115": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110115,

 "regionName": "大兴区"

 },

 "110116": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110116,

 "regionName": "怀柔区"

 },

 "110117": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110117,

 "regionName": "平谷区"

 },

 "110118": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110118,

 "regionName": "密云区"

 },

 "110119": {

 "areaId": 0,

 "children": {},

 "regionGrade": 3,

 "regionId": 110119,

 "regionName": "延庆区"

 }

 },

 "regionGrade": 2,

 "regionId": 110100,

 "regionName": "北京市"

 }

 },

 "regionGrade": 1,

 "regionId": 110000,

 "regionName": "北京市"

 }

}

## 4.2 修改报关信息接口

### 4.2.1 应用场景

当订单商品为跨境商品时，需要报关审核，报关失败需要调用此接口修改报关信息

### 4.2.2 注意事项

1.正常订单请不要调此接口，报关异常的订单才需要调此接口；

### 4.2.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 |  /api/v2/order/reDeclare |
| 功能描述 | 当订单商品为跨境商品时，需要报关审核，报关失败需要调用此接口修改报关信息 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| orderId | 订单ID | string | 必填；与orderSn二选一 |
| orderSn | 订单编号 | string | 必填；与orderId二选一 |
| memberId | 用户ID | string | 必填 |
| address | 收货人地址 | string | 必填 |
| identification | 收货人身份证号 | string | 必填 |
| mobile | 收货人手机号 | string | 必填 |
| area | 区域ID | number | 必填 |
| name | 收货人姓名 | string | 必填 |
| emblumside | 身份证反面照片 | string | 选填； 异常类型是"未上传身份证照片"时必传 |
| photoside | 身份证正面照片 | string | 选填； 异常类型是"未上传身份证照片"时必传 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| data |  | string | @mock="修改报关信息成功" |
| message |  | string | @mock="" |
| success |  | boolean | @mock=true |
| code | 错误码 | string | @mock="" |
| result |  | number | @mock=1 |

### 4.2.4 返回json串 示例

{
    "code": "",
    "data": "修改报关信息成功",
    "message": "",
    "result": 1,
    "success": true
}

## 4.3 申请售后接口

### 4.3.1 应用场景

当遇到需要退货，退款，补发等售后问题时调用此接口

### 4.3.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/order/apply |
| 功能描述 | 当遇到需要退货，退款，补发等售后问题时调用此接口 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | string | 必填 |
| returnDes | 描述 | string | 必填 |
| applyMoney | 申请金额 | number | 必填 |
| applyReason | 申请原因 | string | 必填 |
| photos | 图片 | string | isGoodsReceived 非0时，必填 |
| orderSn | 订单编号 | string | 必填 |
| phone | 联系方式 | string | 选填 |
| contact | 联系人 | string | 选填 |
| isGoodsReceived | 是否收到商品 0:未收到货 1:已收到货 | number | 选填 |
| type | 售后类型 | number | 必填，5：仅退款，1：退货且退款 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| message |  | string | @mock="" |
| result |  | number | @mock=1 |
| success |  | boolean | @mock=true |
| code | 错误码 | string | @mock="" |
| data |  | string | @mock=申请成功 |

### 4.3.3 返回json串 示例

{
    "code": "",
    "data": "申请成功",
    "message": "",
    "result": 1,
    "success": true
}

## 4.4 确认收货接口

### 4.4.1 应用场景

确认收货后，订单被置为已完成。（无法进行售后，退款等操作）

### 4.4.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/order/rogConfirm |
| 功能描述 | 确认收货后，订单被置为已完成。（无法进行售后，退款等操作） |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 会员ID | string | 必填; |
| orderIds | 订单IDs | string | 必填；与orderSns二选一  |
| orderSns | 订单sns | string | 必填；与orderIds二选一 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code | 错误码 | string | @mock="" |
| data |  | string | @mock="确认收货成功" |
| message |  | string | @mock="" |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 |

### 4.4.3 返回json串 示例

{
    "code": "",
    "data": "确认收货成功",
    "message": "",
    "result": 1,
    "success": true
}

## 4.5 获取快递信息接口

### 4.5.1 应用场景

订单发货后可通过此接口获取到物流信息

### 4.5.2 注意事项

1.按需调用，禁止轮询。

### 4.5.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | api/v2/order/orderKuaidi |
| 功能描述 | 订单发货后可通过此接口获取到物流信息 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| orderSn | 订单编号 | string | 必填；校验参数，防止随意调用 @mock=201706309456390 |
| dlyCode | 快递代码 | string | 必填；@mock=zhongtong |
| shipNo | 运单号 | string | 必填；@mock=535260833968 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| message |  | string | @mock= |
| data |  | object |  |
| comName | ---快递公司名称（中文） | string | @mock=圆通快递 |
| state | ---快递查询状态 | string | @mock=3 |
| data|13 | --- | array<object> |  |
| context | ------物流内容 | string | @mock=$order('客户 签收人: 梁爽 已签收','浙江省杭州市文三路公司派件人: 张炜梁 派件中 派件员电话','杭州转运中心公司 已发出','快件到达 浙江省杭州市文三路公司','杭州转运中心公司 已发出,下一站 浙江省杭州市文三路','快件到达 杭州转运中心公司','临海转运中心公司 已发出','快件到达 临海转运中心公司','浙江省台州市椒江区公司 已发出,下一站','浙江省台州市黄岩区公司 已发出,下一站','浙江省台州市椒江区公司 已发出,下一站 临海转运中心','浙江省台州市椒江区公司 已打包,发往下一站 杭州转运中心','浙江省台州市椒江区公司 已揽收') |
| time | ------物流时间 | string | @mock=$order('2015-05-18 11:39:03','2015-05-18 08:16:48','2015-05-18 07:38:26','2015-05-18 07:04:35','2015-05-18 05:12:56','2015-05-18 03:13:52','2015-05-18 00:06:20','2015-05-17 20:57:21','2015-05-17 20:38:10','2015-05-17 20:05:34','2015-05-17 18:46:55','2015-05-17 18:25:29','2015-05-17 18:19:43') |
| codenumber | ---快递单号 | string | @mock=880132606034451359 |
| message | --- | string | @mock=ok |
| updateTime | ---更新时间 | string | @mock=2015-05-19 15:30:47 |
| com | ---快递公司代码 | string | @mock=yuantong |
| result |  | number | @mock=1 |
| code | 错误码 | string | @mock= |
| success |  | boolean | @mock=true |

### 4.5.4 返回json串 示例

{
    "code": "",
    "data":
    {
        "codenumber": "880132606034451359",
        "com": "yuantong",
        "comName": "圆通快递",
        "data": [
        {
            "context": "客户 签收人: 梁爽 已签收",
            "time": "2015-05-18 11:39:03"
        }
        ,
        {
            "context": "浙江省杭州市文三路公司派件人: 张炜梁 派件中 派件员电话",
            "time": "2015-05-18 08:16:48"
        }
        ,
        {
            "context": "杭州转运中心公司 已发出",
            "time": "2015-05-18 07:38:26"
        }
        ,
        {
            "context": "快件到达 浙江省杭州市文三路公司",
            "time": "2015-05-18 07:04:35"
        }

        ],
        "message": "ok",
        "state": "3",
        "updateTime": "2015-05-19 15:30:47"
    }
    ,
    "message": "",
    "result": 1,
    "success": true
}

## 4.6 取消订单

### 4.6.1 应用场景

订单未付款前可以取消订单

### 4.6.2 注意事项

1.订单一旦付款成功，无法取消订单；

### 4.6.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | api/v2/order/orderCancel |
| 功能描述 | 订单未付款前可以取消订单 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| orderIds | 多订单ID逗号隔开 | string | 必填; 与orderSns二选一; |
| orderSns | 多订单号逗号隔开 | string | 必填; 与orderIds二选一 |
| memberId | 会员ID | string | 必填  |
| reason | 取消原因 | string | 选填; |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| data |  | string | @mock=“取消订单成功” |
| code | 错误码 | string | @mock="" |
| result |  | number | @mock=1 |
| message |  | string | @mock="" |
| success |  | boolean | @mock=true |

### 4.6.4 返回json串 示例

{
    "code": "",
    "data": "“取消订单成功”",
    "message": "",
    "result": 1,
    "success": true
}

## 4.7 取消申请售后接口

### 4.7.1 应用场景

取消申请售后

### 4.7.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | api/v2/order/cancelApply |
| 功能描述 | 取消申请售后 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| orderSn | 订单号 | string | 必填 |
| memberId | 会员ID | string | 必填 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| result |  | number | @mock=1 |
| data |  | string | @mock=取消申请成功 |
| message |  | string | @mock= |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |

### 4.7.3 返回json串 示例

{
    "code": "",
    "data": "取消申请成功",
    "message": "",
    "result": 1,
    "success": true
}

## 4.8 发起余额扣款接口

### 4.8.1 应用场景

三方平台对接时使用海带账户的余额扣款，完成订单付款

### 4.8.2 注意事项

1. 余额扣款失败可能是因为余额不足的原因。
2. 禁止轮询调用。

### 4.8.3 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 |  /api/v2/finance/toPayForBalanceWithoutCode |
| 功能描述 | 发起余额扣款，不需要验证码 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | string | 必填； |
| orderIds | 订单ID | string | 必填；ID/SN二选一 以逗号分隔 |
| orderSns | 订单SN | string | 必填；SN/ID 二选一 以逗号分隔 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code | 错误码 | string |  |
| data |  | object |  |
| tradeNo | 交易单号 | string |  |
| message | 错误信息 | string |  |
| result | 返回结果 | number | 1=成功，0=失败 |
| success |  | boolean | true |

### 4.8.4 返回json串 示例

{

 "code": "",

 "data":{}

 ,

 "message": "",

 "result": 1,

 "success": true

}

## 4.9 交易资金流水列表

### 4.9.1 应用场景

三方平台对接时获取在海带账号交易流水情况

### 4.9.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/member/dealRecordList |
| 功能描述 | 交易记录 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| pageNum | 页码号 | number | 必填 |
| exchangeType | 交易方式，（0全部12余额1支付宝6微信4银行卡9线下转账） | number | 必填 @mock=1 |
| memberId | 用户ID | string | 必填 @mock=985 |
| pageSize |  |  | 必填 |
| businessType | 交易类型，（0全部5购买4分拥1充值2提现7活动奖励3退款） | number | 必填 @mock=1 |
| startDate | 开始日期 | string | 选填 @mock=2017-06-14 |
| endDate | 结束日期 | string | 选填 @mock=2017-06-13 |
| orderSn | 订单号 | string | 选填 @mock=123123123 |
| businessStatus | 交易状态，（不传全部0进行中1成功2失败） | number | 选填 @mock=1 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 |
| data |  | object |  |
| result|2 | --- | array<object> |  |
| statusName | ------状态 | string | @mock=$order('已付款','已付款') |
| creationTime | ------交易时间 | string | @mock=$order('2017-04-20 09:37:36','2017-04-20 09:37:36') |
| bussinessName | ------交易类型 | string | @mock=$order('购买','购买') |
| dealModeStr | ------处理方式 | string | @mock=$order('余额','余额') |
| dealRecordNo | ------订单 | string | @mock=$order('20170609125360','20170609125360') |
| amount | ------金额 | number | @mock=$order(100,100) |
| sn | ------sn | string | @mock=$order('201505190002090','201505190002090') |
| pageSize | --- | number | @mock=10 |
| totalCount | --- | number | @mock=10 |
| currentPageNo | --- | number | @mock=1 |
| totalPageCount | --- | number | @mock=1 |
| message |  | string | @mock= |

### 4.9.3 返回json串 示例

{

 "code": "",

 "data":

 {

 "currentPageNo": 1,

 "pageSize": 10,

 "result": [

 {

 "amount": 100,

 "bussinessName": "购买",

 "creationTime": "2017-04-20 09:37:36",

 "dealModeStr": "余额",

 "dealRecordNo": "20170609125360",

 "sn": "201505190002090",

 "statusName": "已付款"

 }

 ],

 "totalCount": 10,

 "totalPageCount": 1

 }

 ,

 "message": "",

 "result": 1,

 "success": true

}

## 5.0 获取保税订单支付二维码

### 5.0.1 应用场景

保税订单下单后，获取支付二维码信息

### 5.0.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/order/getPayQrcode |
| 功能描述 | 获取支付二维码 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | string | 必填 @mock=985 |
| orderSn | 订单号 | string | 必填 @mock=123123123 |
| paymentId | 扫码付款方式ID，（2支付宝13微信[名称海带] 14微信[名称母婴商户联盟]） | number | 必填 @mock=2 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 |
| data |  | object |  |
| qrcode | 二维码信息 | String | @mock=10 |
| qrcodeImg | 二维码图片地址 | String | @mock=10 |
| payMoney | 支付金额 | number | @mock=1 |
| invalidTime | 二维码失效时间 | Date | @mock=1 |
| message |  | string | @mock= |

### 5.0.3 返回json串 示例

**{**

**"code": "",**

**"data": {**

**"invalidTime": "2019-01-22 12:56:10",**

**"payMoney": 988,**

**"qrcode": "weixin://wxpay/bizpayurl?pr=hd0Z0RZ",**

**"qrcodeImg": "<http://img.dev.seatent.com/statics/c5f311b0de8b14fec801464ca453a410/images/payQrcode/8293b5fde6c84d50ba01485ebb014351/wxpay-201901220517356036.png>"**

**},**

**"message": "",**

**"result": 1,**

**"success": true**

**}**

## 5.1获取支付方式列表

### 5.1.1 应用场景

获取系统支付的支付方式列表

### 5.1.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/finance/getThirdPayChannels |
| 功能描述 | 获取支付方式列表 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | String | 必填 @mock=985 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 |
| data |  | object |  |
| dataType | 发起支付返回数据格式 | String | @mock=qrcode|json |
| name | 支付方式名称 | String | @mock=支付宝扫码支付 |
| paymentId | 支付方式ID | number | @mock=2 |
| type | 支付方式代码 | String | @mock=alipayQrcode |
| message |  | string | @mock= |

### 5.1.3 返回json串 示例

{

 "code": "",

 "data": [

 {

 "dataType": "qrcode",

 "name": "支付宝扫码支付",

 "paymentId": 2,

 "type": "alipayQrCode"

 }

 ],

 "message": "",

 "result": 1,

 "success": true

}

## 5.2发起支付

### 5.2.1 应用场景

用户发起支付获取支付信息

### 5.2.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/finance/requestPay |
| 功能描述 | 发起支付 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | String | 必填 |
| orderIds | 订单IDS, 多个订单以英文逗号隔开 保税订单不支持传多个 | String | 必填 |
| orderType | 固定值 传1 | number | 必填 @mock=1 |
| paymentId | 支付方式ID | number | 必填 @mock=2 |
| v | 版本号，固定v2 | String | 必填 @mock=v2 |
| accountId | 账号ID | String | 必填 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 |
| data |  | object |  |
| dataType | 发起支付返回数据格式 | String | @mock=qrcode|json |
| payMoney | 支付金额 | number | @mock=支付宝扫码支付 |
| payOrderNo | 支付单号 | String |  |
| codeUrl | 支付方式ID | String | 该参数只有dataType=qrcode才返回 |
| payInfo | 支付信息<微信|支付宝支付信息> | Map | 该参数只有dataType=json才返回 |
| message |  | string | @mock= |

### 5.2.3 返回json串 示例

{

 "code": "",

 "data": {

 "payMoney": 1575,

 "dataType": "qrcode",

 "codeUrl": "https://qr.alipay.com/bax01514rdjp26bdsxs9808e",

 "shipName": "小黄人",

 "payOrderNo": "201905140465813250"

 },

 "message": "",

 "result": 1,

 "success": true

}

## 5.3查询支付结果

### 5.3.1 应用场景

发起支付后查询支付状态

### 5.3.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/finance/queryPayStatus |
| 功能描述 | 查询支付状态 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | String | 必填 |
| payOrderNo | 支付单号，发起支付返回的 | String | 必填 |
| paymentId | 发起支付方式ID | number | 必填 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 |
| data | 1支付成功 0未支付 | number |  |
| message |  | string | @mock= |

### 5.3.3 返回json串 示例

{

 "code": "",

 "data": 0,

 "message": "",

 "result": 1,

 "success": true

}

## 5.4保存支付原始信息

### 5.4.1 应用场景

保税商品使用支付宝支付后，把支付原始信息保存下来以便海关查验

### 5.4.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/finance/savePaymentInfo |
| 功能描述 | 保存支付原始信息 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | String | 必填 |
| payOrderNo | 支付单号，发起支付返回的 | String | 必填 |
| tradeNo | 支付流水号 | String | 必填 |
| paymentId | 发起支付方式ID | number | 必填，默认一个值 |
| callbackId | 支付回调ID | number | 选填 |
| payMoney | 支付金额 | BigDecimal | 必填 （元） |
| customsCode | 海关代码 | String | 选填 |
| companyCode | 支付公司代码 | String | 必填，默认ALIPAY |
| sendPayInfo | 发送支付报文 | String | 必填 |
| receivePayInfo | 同步响应报文 | String | 必填 |
| noticePayInfo | 异步通知报文 | String | 选填 |
| status | 状态 | number | 0.不能报关，1.报关成功，2.等待报关，3，报关中，-1报关失败 |
| taxAmount | 税费 | BigDecimal | 必填（元） |
| paySuccessTime | 支付成功时间 | String | 必填 yyyy-MM-dd HH:mm:ss |
| sourceType | 来源类型 | number | 必填，默认写3 |
| config | 支付配置 | String | 必填 |
| accountNo | 收款账号 | String | 必填 |
| accountName | 收款账号名称 | String | 必填 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 成功，0 失败 |
| data |  | number | 操作成功 |
| message |  | string | @mock= |

### 5.3.3 返回json串 示例

{

 "code": "",

 "data": 操作成功,

 "message": "",

 "result": 1,

 "success": true

}

## 5.5 修改\上传身份证照片

### 5.5.1 应用场景

方便直邮商品需要上传身份证照片

### 5.5.2 API内容

基本信息：

|  |  |
| --- | --- |
| API名称 | /api/v2/member/editMemberAddress |
| 功能描述 | 修改用户地址 |

输入参数：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| memberId | 用户ID | String | 必填 |
| accountId | 用户账号id | String | 必填 |
| provinceId | 省id | String | 必填 |
| cityId | 市id | String | 必填 |
| regionId | 区id | String | 必填 |
| photoSide | 身份证正面 | String | 选填 |
| emblumSide | 身份证反面 | String | 选填 |
| addr | 详细地址 | String | 必填 |
| id | 地址id | String | 必填（订单列表或者详情里面获取addressId） |
| mobile | 手机号码 | String | 必填 |
| name | 姓名 | String | 必填 |
| idNo | 身份证号码 | number | 必填 |

返回结果：

| **变量名** | **含义** | **类型** | **备注** |
| --- | --- | --- | --- |
| code |  | string | @mock= |
| success |  | boolean | @mock=true |
| result |  | number | @mock=1 成功，0 失败 |
| data |  | number | 操作成功 |
| message |  | string | @mock= |

### 5.5.3 返回json串 示例

**{**

"code": "",

"data": **null**,

"message": "更新状态成功",

"result": **1**,

"success": **true**

**}**