

# 品优购 API 文档

注意：

开发环境的域名为：暂无

生产环境的域名为： <https://www.uinav.com/wx/>

说明：

在所有的 url 中，凡是带有 my 的，服务器都是需要验证 token 的

## 一、首页

### 1.1 、获取轮播图

地址	api/public/v1/home/swiperdata
应用场景	
作用描述	获取首页轮播图
请求方式	GET
传入 api 的参数	无

返回数据格式	JSON
返回数据格式样例	<pre>{   - message: [     - {       image_src: "http://p61myfkof.bkt.clouddn.com/banner1.png",       open_type: "navigate",       navigator_url: "/pages/goods_detail?goods_id=140"     },     - {       image_src: "http://p61myfkof.bkt.clouddn.com/banner2.png",       open_type: "navigate",       navigator_url: "/pages/goods_detail?goods_id=395"     },     - {       image_src: "http://p61myfkof.bkt.clouddn.com/banner3.png",       open_type: "navigate",       navigator_url: "/pages/goods_detail?goods_id=38"     }   ],   - meta: {     msg: "获取成功",     status: 200   } }</pre>
服务器返回参数说明	image_src: 图片地址 open_type: 导航方式, 支持 navigate navigator_url: 导航地址

## 1.2 、获取分类列表

地址	api/public/v1/home/catitems
应用场景	<div>分类</div> <div>秒杀拍</div> <div>超市购</div> <div>母婴品</div>
作用描述	首页获取分类列表数据

请求方式	GET
传入 api 的参数	无
返回数据格式	JSON
返回数据格式样例	<pre> {   - message: [     - {       name: "分类",       image_src: "http://p6lmfkof.bkt.clouddn.com/icon_index_nav_4@2x.png",       open_type: "switchTab",       navigator_url: "/pages/category"     },     - {       name: "秒杀拍",       image_src: "http://p6lmfkof.bkt.clouddn.com/icon_index_nav_3@2x.png",       open_type: "navigate",       navigator_url: "/pages/goods_list?query=秒杀"     },     - {       name: "超市购",       image_src: "http://p6lmfkof.bkt.clouddn.com/icon_index_nav_2@2x.png",       open_type: "navigate",       navigator_url: "/pages/goods_list?query=超市"     },     - {       name: "母婴品",       image_src: "http://p6lmfkof.bkt.clouddn.com/icon_index_nav_1@2x.png",       open_type: "navigate",       navigator_url: "/pages/goods_list?query=母婴"     }   ],   - meta: {     msg: "获取成功",     status: 200   } } </pre>
服务器返回参数说明	name:分类名称 img_src:分类图标地址 open_type:打开方式 navigator_url:导航 URL 地址

### 1.3 、获取楼层数据

地址	api/public/v1/home/floordata
----	------------------------------

应用场景	
作用描述	首页获取楼层数据
请求方式	GET
传入 api 的参数	无
返回数据格式	JSON
返回数据格式样例	<pre>{   - message: [     - {       - floor_title: {         name: "时尚女装",         image_src: "http://p61myfkof.bkt.clouddn.com/pic_floor01_title.png"       },       - product_list: [         - {           name: "优质服饰",           image_src: "http://p61myfkof.bkt.clouddn.com/pic_floor01_1@2x.png",           image_width: "232",           open_type: "navigate",           navigator_url: "/pages/goods_list?query=服饰"         },         + {...},         + {...},         + {...},         + {...}       ]     },     + {...},     + {...}   ],   - meta: {     msg: "获取成功",     status: 200   } }</pre>

服务器返回参数说明	floor_title.name:一级楼层标题 floor_title.image_src:一级楼层图片地址 product_list[0].name:二级楼层名称 product_list[0].image_src:二级楼层图片地址 product_list[0].image_width:二级楼层图片宽度 product_list[0].open_type:二级楼层导航方式 product_list[0].navigator_url:二级楼层导航 url 地址
-----------	---

## 二、商品详情

### 2.1 、获取搜索列表数据

地址	api/public/v1/goods/search
应用场景	<div><div><div>紫米ZMI 小米QC3.0快充车充</div><div>5V2.4A双USB智能输出 一拖二手...</div><div>¥69.00</div></div></div>
作用描述	根据关键字搜索数据
请求方式	GET
传入 api 的参数	query : 搜索关键字  pagesize: 页容量  pagenum: 当前页
返回数据格式	JSON

返回数据格式样例	<pre>{   - message: {     total: 13,     pagenum: "1",     - goods: [       - {         goods_id: 542,         cat_id: 516,         goods_name: "小米 (MI) 米家压力IH电饭煲 智能烹饪 压力IH电磁环绕加热 手机智能预约 3L黄金容量 灰铸铁内胆",         goods_price: 899,         goods_number: 100,         goods_weight: 100,         goods_big_logo: "",         goods_small_logo: "http://image4.suning.cn/uimg/b2c/newcatentries/0000000000-000000000152287756_2_200x200.jpg",         add_time: 1514257949,         upd_time: 1514257949,         hot_number: 0,         is_promote: false,         cat_one_id: 483,         cat_two_id: 487,         cat_three_id: 516       },       - {</pre>
服务器返回参数说明	

## 2.2 、获取商品详情数据


地址	api/public/v1/goods/detail?goods_id=140
应用场景	
作用描述	根据商品 id 获取该商品的详细信息
请求方式	GET
传入 api 的参数	goods_id : 商品 id

返回数据格式	JSON
返回数据格式样例	<pre>{   - message: {     goods_id: 140,     cat_id: 134,     goods_name: "【海外购自营】红印 Red Seal 薄荷茶 25包/盒 新西兰进口 舒心畅饮 茶叶",     goods_price: 39,     goods_number: 100,     goods_weight: 100,     goods_introduce: "&lt;div class='lazyimg'&gt;&lt;p&gt;&lt;a href='https://g.suning.com/' target='_blank'&gt;&lt;img data-src='https://from=mobile&amp;format=80q.webp' alt='' src='https://image.suning.cn/uimg/sop/commodity/302227931152378466201300.&lt;p&gt;&lt;img data-src='https://image.suning.cn/uimg/sop/commodity/849180101203166897581500_x.jpg?from=mobile&amp;format=80q.webp' width=     goods_big_logo: "",     goods_small_logo: "http://image5.suning.cn/uimg/b2c/newcatentries/0000000000-000000000695599644_1_200x200.jpg",     goods_state: 2,     is_del: "0",     add_time: 1514256374,     upd_time: 1514256374,     delete_time: null,     hot_number: 0,     is_promote: false,     cat_one_id: 70,     cat_two_id: 115,     cat_three_id: 134,     goods_cat: "70,115,134",     + pics: [...],     + attrs: [...]   },   - meta: {     msg: "获取成功",     status: 200   } }</pre>
服务器返回参数说明	goods_id:商品 id goods_name:商品名称 goods_price:商品价格 goods_introduce:商品介绍 pics:轮播图 attrs:商品属性

## 三、分类

### 3.1 、查询分类数据

地址	api/public/v1/categories
----	--------------------------

应用场景	
作用描述	获取分类数据
请求方式	GET
传入 api 的参数	无
返回数据格式	JSON
返回数据格式样例	<pre>{   - message: [     - {       cat_id: 1,       cat_name: "大家电",       cat_pid: 0,       cat_level: 0,       cat_deleted: false,       cat_icon: "",       + children: [...]     },     + {...},     + {...},     + {...},     + {...},     + {...},   ] }</pre>
服务器返回参数说明	cat_id:分类 id cat_name:分类名称 cat_level:分类级别 cat_icon: 分类图片







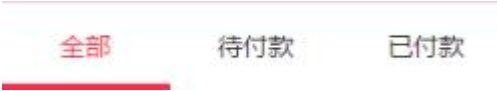
传入 api 的参数	<p><b>code</b> 用户登录凭证 通过 wx.login()获取</p> <p><b>encryptedData</b> 完整用户信息密文 通过 wx.getUserInfo ()获取</p> <p><b>iv</b> 加密算法的初始向量 通过 wx.getUserInfo ()获取</p> <p><b>rawData</b> 用户信息原始数据字符串 通过 getUserInfo 获取, JSON.stringify() 转字符串</p> <p><b>signature</b> 使用 sha1 得到字符串 通过 getUserInfo 获取</p>
返回数据格式	JSON
返回数据格式样例	<pre>{   "message": {     "user_id": 13,     "user_email_code": null,     "is_active": null,     "user_sex": "男",     "user_qq": "",     "user_tel": "",     "user_xueli": "本科",     "user_hobby": "",     "user_introduce": null,     "create_time": 1527951108,     "update_time": 1527951108,     "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWQiOiJlZCJpYXQiOiJlMjg3Nzg4NzIsImV4cCI6MTUyODg2NTI3Mn0.7gkt51lh6CYRNOglDnj"   },   "meta": {     "msg": "登录成功",     "status": 200   } }</pre>
服务器返回参数说明	<p><b>user_id</b>: 用户 id</p> <p><b>token</b>: 登录成功后, 后台给你返回的令牌</p>

## 5.2 、下单操作

地址	api/public/v1/my/orders/create
应用场景	创建订单
作用描述	登录之后, 创建订单
请求方式	POST
传入 api 的参数	<p><b>请求头:</b></p> <p><b>"Authorization" : token</b> //需要设置 token 带给后台</p>

	<p>请求体:</p> <p><code>order_price</code> 订单价格</p> <p><code>consignee_addr</code> 订单地址</p> <p><code>goods</code> 商品列表内部存放商品对象 (<code>goods_id</code>, <code>goods_number</code> 和 <code>goods_price</code>)</p>
返回数据格式	JSON
返回数据格式样例	<pre>{   "message": {     "order_id": 80,     "user_id": 13,     "order_number": "HMDD20180612000000000080",     "order_price": 1297,     "order_pay": "0",     "is_send": "否",     "trade_no": "",     "order_fapiao_title": "个人",     "order_fapiao_company": "",     "order_fapiao_content": "",     "consignee_addr": "广东省广州市海珠区新港中路397号",     "pay_status": "0",     "create_time": 1528779631,     "update_time": 1528779631,     "order_detail": null,     "goods": [{       "id": 105,       "order_id": 80,       "goods_id": 38,       "goods_price": 98,       "goods_number": 1,       "goods_total_price": 98     }, {</pre>
服务器返回参数说明	<p><code>order_id</code>: 订单 id</p> <p><code>user_id</code>: 用户 id</p> <p><code>order_number</code>: 订单号，支付及查询订单状态时候需要用到</p> <p><code>order_price</code>: 订单价格</p> <p><code>consignee_addr</code>: 收货地址</p> <p><code>goods</code>: 下单时，购买商品的数组</p>

### 5.3 、查询订单支付状态

地址	api/public/v1/my/orders/chkOrder
应用场景	
作用描述	获取某笔订单的支付状态
请求方式	POST
传入 api 的参数	请求头:  "Authorization": token //需要设置 token 带给后台  请求体: order_number: 订单号
返回数据格式	JSON
返回数据格式样例	<pre>{   "message": "支付成功",   "meta": {     "msg": "验证成功",     "status": 200   } }</pre>
服务器返回参数说明	message: 支付状态消息

### 5.4 、查询订单列表

地址	api/public/v1/my/orders/all?type=1
----	------------------------------------

应用场景	
作用描述	查询订单
请求方式	GET
传入 api 的参数	<p>请求头:</p> <p>"Authorization": token //需要设置 token 带给后台</p> <p>type: 1 所有 2 未支付 3 已支付 4 退款/退货</p>
返回数据格式	JSON

返回数据格式样例	<pre>{   "message": {     "count": 5,     "orders": [{       "order_id": 80,       "user_id": 13,       "order_number": "HMDD20180612000000000080",       "order_price": 1297,       "order_pay": "0",       "is_send": "否",       "trade_no": "",       "order_fapiao_title": "个人",       "order_fapiao_company": "",       "order_fapiao_content": "",       "consignee_addr": "广东省广州市海珠区新港中路397号",       "pay_status": "0",       "create_time": 1528779631,       "update_time": 1528779631,       "order_detail": null,       "goods": [{</pre>
服务器返回参数说明	<p>orders[0].order_id: 订单 id</p> <p>orders[0].user_id: 用户 id</p> <p>orders[0].order_number: 订单号</p> <p>orders[0].order_price: 订单价格</p> <p>orders[0].pay_status: 支付状态 0: 未支付 1: 已支付</p> <p>orders[0].consignee_addr: 收货地址</p> <p>orders[0].goods: 下单时, 购买商品的数组</p> <p>orders[0].total_count: 订单总件数</p> <p>orders[0].total_price: 订单总金额</p>

## 六、支付

### 6.1 、生成预支付交易单

地址	api/public/v1/my/orders/req_unifiedorder
----	--

应用场景	<div><p>微信支付</p><p>请用开发者本人微信（黄Sir）扫描以上二维码进行支付调试</p></div>
作用描述	生成预支付交易单
请求方式	POST
传入 api 的参数	<p>请求头:</p> <p>"Authorization": token //需要设置 token 带给后台</p> <p>请求体:</p> <p>order_number: 订单号</p>
返回数据格式	JSON
返回数据格式样例	<pre>{   "message": {     "wxorder": {       "timeStamp": "1528794423",       "nonceStr": "ewsWKtBiKX42X58A",       "package": "prepay_id=wx121707030376297e21f53e691011696857",       "signType": "MD5",       "paySign": "84557DBAC3EF0660697653DF61E2B9D4"     },     "order_number": "HMDD20180612000000000080"   },   "meta": {     "msg": "预付订单生成成功",     "status": 200   } }</pre>



服务器返回参数说明	<code>wxorder.timeStamp</code> :时间戳 <code>wxorder.nonceStr</code> :随机字符串 <code>wxorder.package</code> :包含有 <code>prepay_id</code> <code>wxorder.signType</code> :签名类型 <code>wxorder.paySign</code> :签名字符串 <code>order_number</code> :订单号
-----------	---