

通淘供应链API接口文档描述了对外提供的接口与协议，包括订单、商品、库存等接口定义；

通淘供应链API接口文档

深圳市通拓科技有限公司

龙岗区华南城1号交易广场5F

0755-83998006

2017年4月18日

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修改日期 | 修改人 | 修改说明 |
| 0.1.0 | 2016-08-29 |  | 接口定义初稿 |

# 接口定义

## 采购订单

### 查询采购订单

|  |  |  |  |
| --- | --- | --- | --- |
| 请求名称 | 查询采购订单 | | |
| 请求方式 | POST | URL | openapi/purchase/order/list |
| 传入参数 | | | |
| 参数名 | 可为空 | 参数类型 | 描述 |
| status | Y | String | 可选。状态码 "WAIT\_PAY":(未付款) "CANCEL":(已取消) "PAY\_SUCCESS": (已付款)INVALID："已失效" |
| currPage | Y | Integer | 可选。页码，不传时，默认为1 |
| pageSize | Y | Integer | 可选。页记录数，不传时，默认为10,最大值为100 |
| title | Y | String | 模糊搜索（采购单号，商品名称） |
| sorderDate | Y | String | 下单时间 开始（2016-08-10） |
| eorderDate | Y | String | 下单时间 结束（2016-08-20） |
| spaydate | Y | String | 支付时间 开始（2016-08-10） |
| epaydate | Y | String | 支付时间 结束（2016-08-20） |
| ltc | N | String | 登录令牌（登录成功时返回的ltc\_key） |
| 请求示例 | {  "ltc": "75f138f0-2484-4c7d-b138-f024843c7d73",  "pageSize": 10,  "pageCount": 1,  "status": "PAY\_SUCCESS",  "titile": "",  "sorderDate": "2016-08-10",  "eorderDate ": "2016-08-20"  } | | |
| 响应示例 | 成功：  {  "code": 100,  "response": {  "currPage": 1,  "pageSize": 10,  "rows": 1,  "totalPage": 1,  "hasnext": false,  "data\_list": [  {  "id": 2055,  "email": "18206698184",  "purchaseOrderNo": "CG201608110000002448",  "status": 1,  "purchaseDate": 1470881722854,  "sorderDate": "2016-08-11 10:15:22",  "statusMes": "已付款",  "purchaseTotalAmount": 9.46,  "payDate": 1470881746474,  "spayDate": "2016-08-11 10:15:46",  "purchaseDiscountAmount": 9.46,  "details": [  {  "productImg": "http://static.tomtop.com.cn/imaging/imaging/product/IF639/IF639-1-3a9d-jlqn.jpg",  "productName": "麦蒂斯金装啤酒500ml",  "sku": "IF639",  "purchasePrice": 9.46,  "marketPrice": 17.85,  "qty": 1,  "weight": 1,  "purchaseId": 2055,  "warehouseId": 2024,  "warehouseName": "深圳仓",  "totalPrices": 9.46,  "afterDiscountPrice": 9.46,  "disProfitMargin": 0,  "discountProfitMargin": 0,  "purchaseCostPrice": 0,  "disProfit": 1.69,  "disStockFee": 1.1,  "disShippingType": "X2",  "disOtherCost": 0.65,  "disTransferFee": 0,  "dislistFee": 0,  "distradeFee": 0,  "dispayFee": 0,  "dispostalFee": 0,  "disimportTar": 0,  "disgst": 0,  "disinsurance": 0,  "distotalvat": 0,  "disCifPrice": 2.19,  "cost": 2.19,  "disFreight": 0,  "disStockId": null,  "disPrice": 5.63,  "categoryId": null,  "isgift": false  }  ],  "sid": 1682,  "discount": null,  "salesAmount": null,  "purchaseType": 2,  "cancelDate": null,  "cancelDateStr": null,  "bbcPostage": 6,  "logisticsMode": "中通快递-国内",  "couponsCode": "",  "couponsAmount": 0,  "deductionAmount": null,  "isPro": false,  "isChoose": false  }  ]  }  }  失败：  {  "code": 101,  "msg": "最大页数不能超过100"  }  100 正常  101 参数错误  110 查询异常 | | |

### 下采购订单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 请求名称 | 下采购订单 | | | |
| 请求方式 | POST | URL | openapi/purchase/order |
| 传入参数 | | | | |
| 参数名 | 可为空 | 参数类型 | 描述 | |
| details | N | List | 包含商品(sku,wareId,qty)的集合 | |
| sku | N | String | 必传。商品sku | |
| wareId | N | Integer | 必传。仓库id | |
| isgift | Y | boolean | 可选。是否为赠品，传true则为赠品，false或不则为非赠品 | |
| qty | N | Integer | 必传。每个sku购买的数量 | |
| ltc | N | String | 登录令牌（登录成功时返回的ltc\_key） | |
| 请求示例 | {  "ltc": "75f138f0-2484-4c7d-b138-f024843c7d73",  "details": [  {  "sku": "IM180",  "wareId": 2024,  "qty": 5  },  {  "sku": "IF639",  "wareId": 2024,  "qty": 5,  "isgift": true  }  ]  } | | | |
| 响应示例 | 成功：  {  "code": 100,  "response": {  "orderNo": "CG201608290000002592"  }  }  失败：  {  "code": 104,  "msg": "[IF6319\_2024]商品不存在，无法下单"  }  101 参数错误  102 未查询到商品信息  103 商品编码\_仓库编码 库存不足无法下单  104 某商品 不存在，无法下单  105 下单异常  106 商品不满足起批量 | | | |

// JAVA 代码示例

1.下采购单

Map<String,Object> purOrder = Maps.newHashMap();

purOrder.put("ltc", "32947055-ebea-47c1-9470-55ebea77c189");

List<Map<String,Object>> list = Lists.newArrayList();

Map<String,Object> pro1 = null;

pro1 = Maps.newHashMap();

pro1.put("sku", "IM40");

pro1.put("wareId", "2012");

pro1.put("qty", "4");

list.add(pro1);

pro1 = Maps.newHashMap();

pro1.put("sku", "IF639");

pro1.put("wareId", "2024");

pro1.put("qty", "4");

list.add(pro1);

purOrder.put("details", list);

System.out.println(post(Json.toJson(purOrder).toString(),

"http://tomtopx.com.cn/openapi/purchase/order"));

2.查询采购单

Map<String,Object> purOrderList = Maps.newHashMap();

purOrderList.put("ltc", "32947055-ebea-47c1-9470-55ebea77c189");

System.out.println(post(Json.toJson(purOrderList).toString(),

"http://tomtopx.com.cn/openapi/purchase/order/list"));

//java http工具类

public static String post(String requestBody, String url) {

String responString = "";

PostMethod post = new PostMethod(url);

try {

post.setRequestEntity(new StringRequestEntity(requestBody, "", "utf-8"));

post.setRequestHeader("content-type", "application/json;charset=utf-8");

// 发送http请求

client.executeMethod(post);

// 打印返回的信息

responString = new String(post.getResponseBody(), "UTF-8");

} catch (HttpException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} finally {

// 释放连接

post.releaseConnection();

}

return responString;

}

public static String get(Map<String, String> params, String url) {

String responString = "";

GetMethod get = new GetMethod(url);

HttpClient httpClient = new HttpClient();

List<NameValuePair> nvps = new ArrayList<NameValuePair>();

params.forEach(new BiConsumer<String, String>() {

@Override

public void accept(String t, String u) {

nvps.add(new NameValuePair(t, u));

}

});

NameValuePair[] nvpArr = new NameValuePair[params.size()];

nvpArr = nvps.toArray(nvpArr);

try {

// 参数拼接

get.setQueryString(nvpArr);

get.setRequestHeader("content-type", "application/json;charset=utf-8");

// 发送http请求

httpClient.executeMethod(get);

// 打印返回的信息

responString = new String(get.getResponseBody(), "UTF-8");

} catch (HttpException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} finally {

// 释放连接

get.releaseConnection();

// ((SimpleHttpConnectionManager)httpClient.getHttpConnectionManager()).shutdown();

}

return responString;

}