
Table of Contents

Introduction	1.1
开放数据接口	1.2
序言	1.2.1
数据接口版本更新记录	1.2.1.1
约定	1.2.2
接口请求地址	1.2.2.1
系统级参数	1.2.2.2
算法及过滤特殊字符	1.2.3
签名算法	1.2.3.1
字段内特殊字符过滤说明	1.2.3.2
接口方法	1.2.4
sales.getList 查询销售单信息(根据销售单创建时间)	1.2.4.1
aftersales.getList 查询售后单信息(根据售后单创建时间)	1.2.4.2
iostock.getList 查询出入库明细	1.2.4.3
po.add 新建采购单	1.2.4.4
transfer.add 创建出入单	1.2.4.5
pkg.getList 查询捆绑商品信息	1.2.4.6
stock.getAll 商品库存查询接口	1.2.4.7
stock.getDetailList 仓库库存查询接口	1.2.4.8
po.getList 采购单查询接口	1.2.4.9
invoice.getList 发票列表	1.2.4.10
invoice.update 更新纸质发票的打印信息	1.2.4.11
goods.getList 批量获取商品信息	1.2.4.12
delivery.getList 批量获取发货单信息	1.2.4.13
goods.add 添加商品接口	1.2.4.14
sales.getSalesAmount 获取销售单统计数据	1.2.4.15
purchasereturn.getList 获取采购退货单数据	1.2.4.16
purchasereturn.add 新建采购退货单	1.2.4.17
goods.edit 商品编辑	1.2.4.18
appropriation.add 新建调拨单	1.2.4.19
basicmaterial.add 添加基础物料	1.2.4.20
basicmaterial.edit 编辑基础物料	1.2.4.21
basicmaterial.getList 获取基础物料	1.2.4.22
salesmaterial.add 添加销售物料	1.2.4.23
salesmaterial.edit 编辑销售物料	1.2.4.24
salesmaterial.getList 获取销售物料	1.2.4.25

appropriation.getList 调拨单信息和出入库明细列表	1.2.4.26
transfer.getList 调拨单信息和出入库明细列表	1.2.4.27
仓储接口	1.3
序言	1.3.1
数据接口版本更新记录	1.3.1.1
授权及签名算法	1.3.2
授权	1.3.2.1
签名算法	1.3.2.2
ERP调用WMS接口	1.3.3
调用说明	1.3.3.1
数据结构	1.3.3.2
接口列表	1.3.3.3
入库单创建	1.3.3.3.1
查询入库单状态(同步)	1.3.3.3.2
取消入库单(同步)	1.3.3.3.3
创建出库单	1.3.3.3.4
取消出库单(同步)	1.3.3.3.5
出库单状态查询(同步)	1.3.3.3.6
创建转储单	1.3.3.3.7
取消转储单(同步)	1.3.3.3.8
同步供应商	1.3.3.3.9
创建发货单	1.3.3.3.10
取消发货单(同步)	1.3.3.3.11
添加物料	1.3.3.3.12
更新物料	1.3.3.3.13
退货单创建	1.3.3.3.14
取消退货单(同步)	1.3.3.3.15
WMS调用ERP接口	1.3.4
接口地址	1.3.4.1
接口列表	1.3.4.2
wms.stockin.status_update 入库单状态更新	1.3.4.2.1
wms.stockout.status_update 出库单状态更新	1.3.4.2.2
wms.stockdump.status_update 转储单状态更新	1.3.4.2.3
wms.delivery.status_update 发货单状态更新	1.3.4.2.4
wms.reship.status_update 退货单状态更新	1.3.4.2.5
wms.inventory.add 盘点单添加	1.3.4.2.6
wms.stock.quantity 库存对账	1.3.4.2.7

开放数据和仓储的接口开发文档

为开放数据和仓储的接口调用提供说明.

开放数据接口

为商派ERP开放数据接口的调用提供说明.

数据接口版本更新记录

当前文档版本 **Version 1.0.17**

编辑者	版本号	描述
Danny	V1.0.1	初版
Danny	V1.0.2	增加采购单、出入库明细、出入库单接口
George	V1.0.3	增加“销售单”与“销售明细”关键数据关系说明
George	V1.0.4	1,销售单明细增加“item_type”字段，用于区分“捆绑商品” 增加捆绑商品查询接口，用于解析捆绑商品 2,增加捆绑商品结构
George	V1.0.5	增加签名算法说明
George	V1.0.6	由于存在中文编码问题，修改出入库查询中的出入库类型，由String 修改为number
George	V1.0.7	增加出入库记录中出入库价格并增加对应的说明
George	V1.0.8	增加sign的注意事项，让签名算法更加好理解
George	V1.0.9	1、增加库存查询相关接口 与库存数据相关数据结构 2、增加采购单查询相关接口与采购单数据相关数据结构 3、修改创建出入库明细、创建采购单接口，增加免二次审核控制参数 confirm 4、增加发票相关接口与数据结构
George	V1.0.10	1,修改库存查询接口 ,字段以及方式,注意更新! 2,增加商品查询接口
George	V1.0.10	1,修改库存查询接口 ,字段以及方式,注意更新! 2,增加商品查询接口
Brown	V1.0.11	增加发货单接口
Brown	V1.0.12	增加系统级参数timestamp并对改参数进行有效性验证
Brown	V1.0.13	1、 增加商品添加接口 2、 商品接口跟销售单接口特殊字符过滤
Brown	V1.0.14	1、 增加销售单对账接口 2、 增加采购退货单查询接口 3、 增加各接口特殊字符过滤
Brown	V1.0.15	1、 增加采购退货单创建接口 2、 增加商品编辑接口 3、 增加调拨单创建接口 4、 销售单查询接口优化 ① 增加“会员名”字段 ② “logi_no”字段支持补打物流单产生的多物流单号
wangjianjun	V1.0.16	1、原接口文档整体性归整更新
wangjianjun	V1.0.17	1.接口transfer.add新增类型 Y – 分销入库 Z – 分销出库 2.新增接口 transfer.getList 和appropriation.getList

约定

约定

接口请求地址

<http://www.xxx.com/index.php/openapi/rpc/service/>

注： www.xxx.com 为对应ONex-OMS域名

使用ShopeX 套件的客户，需要登录到 ONex-OMS界面时浏览器中的主域名

系统级参数

- [系统级参数明细表](#)
- [请求方式说明](#)
- [请求方式举例](#)
- [注意](#)

系统级参数明细表

参数名称	是否必要	类型	描述
flag	是	string	访问来源标识(ONex-OMS后台开放数据管理中设置)
sign	是	string	签名, 详见签名算法
method	是	string	接口方法,如获取销售单列表: sales.getList
type	否	string	返回格式, 默认json, 还支持xml
charset	否	string	返回编码, 默认utf-8, 还支持gbk
ver	否	number	版本号, 默认1
timestamp	是	number	Unix时间戳, 到秒

请求方式说明

1. POST方式 请求的所有参数可以使用POST方式
2. 直接URL方式（注意：不是GET参数形式）

[http://接口请求地址/系统级参数名称1/值/系统级参数名称2/值/.../业务级参数1/值/业务级参数2/值/.....](#)
 请求的url分为3个部分，第一部分是接口请求地址，第二部分是系统级参数，第三部分是每个接口的业务级参数，系统级参数和业务级参数都采用 key1/value1/key2/value2 的方式依次拼接在接口请求地址后面。

请求方式举例：

[http://www.xxx.com/index.php/openapi/rpc/service/flag/test/sign/C86732F85599E09D76A6BF45F18D5328/method/sales.getList/type/xml/charset/utf-8/ver/1/start_time/2012-11-11/end_time/2012-11-12/page_no/1/page_size/100](#)

以上**2**种请求方式都兼容

注意

调用具体的接口时，需要在ONex-OMS后台，将对应的接口打开！

访问权限:

销售单:

- 销售单列表(getList) 销售金额(getSalesAmount)

售后单:

- 售后单列表(getList)

出入库明细:

- 出入库明细列表(getList)

采购单:

- 新建采购单(add) 采购单信息(getList)

出入库单:

- 新建出入库单(add)



捆绑商品:

- 捆绑商品列表(getList)

库存查询:

- 商品库存(getAll) 仓库库存(getDetailList)

发票操作:

- 待打印发票列表(getList) 更新发票打印状态(update)

商品:

- 商品列表(getList) 新建商品(add) 修改商品(edit)

发货单:

- 发货单列表(getList)

采购退货单:

- 采购退货单列表(getList) 新建采购退货单(add)

调拨单:

- 新建调拨单(add)

状态: 开启 关闭*

算法及过滤特殊字符

签名算法

签名算法说明

http://www.xxx.com/index.php/openapi/rpc/service/flag/test/sign/C86732F85599E09D76A6BF45F18D5328/method/sales.getList/type/xml/charset=utf-8/ver/1/start_time/2012-11-11/end_time/2012-11-12/page_no/1/page_size/100

- 将所有请求参数（key, value 为一组），对数据结构按照key的升序，重新排序
 - 需要对null值进行过滤
 - 需要将boolean型进行转换为1和0
- 将排序后参数组合为新的字符串（keyvaluekeyvalue 中间无分割符）
- 将生成好的字符串进行MD5加密
- 将新的字符串，统一为大写字符串
- 将生成好的字符串加上“私钥”，形成新的字符串（私钥为ONex-OMS系统里面生成）

接口调用私钥： `OWzAjfxLBiaXWdoCYwrC` `随机生成私钥`

- 再进行一次MD5加密，得到加密后的字符串
- 再统一为大写字符串，得到最终的签名字符串

注意：从“系统级参数”开始到“业务级参数”全部作为请求参数.sign 可以不算入，如果放进去，开始 sign = null ，也会被过滤掉的

字段内特殊字符过滤说明

已经过滤的特殊字符如下

字符名称	处理方式	处理结果
回车符 (\r)	替换	空格 (\x20)
制表符 (\t)	替换	空格 (\x20)
换行符 (\n)	替换	空格 (\x20)
反斜线 (\)	替换	斜线 (/)
英文双引号 (")	替换	中文上双引号 (“)

接口方法

sales.getList 查询销售单信息(根据销售单创建时间)

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

创建一个直接出入库的指令

应用级参数

参数名称	是否必要	类型	示例值
start_time	是	date	2012-12-08 18:50:30
end_time	是	date	2012-12-08 18:50:30
page_no	否	number	1
page_size	否	number	100
shop_id	否	json	json_encode(array('1587965fb4d4b5afe8428a4a024feb0d','1e48c4

返回结果

注意：已经做过特殊字符处理的字段如下：shop_code,shop_name,order_no, member_name, sale_no, pay_method, order_check_op, logi_name, logi_no, branch_name, branch_bn, delivery_no, consignee, consignee_addr, consignee_zip, consignee_tel, consignee_mobile, consignee_email, name, spec_name, item_type

XML数据，例：

成功

```
<response>
  <count>6337</count>
  <lists>
    <shop_code><![CDATA[shopex]]></shop_code>
    <shop_name><![CDATA[shopex]]></shop_name>
```

```

<order_no>194276753441633</order_no>
<member_name><![CDATA[danny]]></member_name>
<sale_no><![CDATA[S20121219000001]]></sale_no>
<pay_method><![CDATA[支付宝]]></pay_method>
<sale_time>2012-12-08 18:50:30</sale_time>
<order_create_time>2012-12-08 18:50:30</order_create_time>
<pay_time>2012-12-08 18:50:30</pay_time>
<ship_time>2012-12-08 18:50:30</ship_time>
<order_check_op><![CDATA[admin]]></order_check_op>
<order_check_time>2012-12-08 18:50:30</order_check_time>
<goods_amount>163.000</goods_amount>
<freight_amount>0.000</freight_amount>
<additional_amount>0.000</additional_amount>
<has_tax><![CDATA[否]]></has_tax>
<pmt_amount>0.000</pmt_amount>
<sale_amount>163.000</sale_amount>
<logi_name><![CDATA[圆通速递]]></logi_name>
<logi_no>7104892994</logi_no>
<branch_name><![CDATA[上海仓]]></branch_name>
<delivery_no>1211101100004</delivery_no>
<consignee><![CDATA[danny]]></consignee>
<consignee_area><![CDATA[上海-上海市-徐汇区]]></consignee_area>
<consignee_addr><![CDATA[上海上海市徐汇区桂林路396号]]></consignee_addr>
<consignee_zip>200233</consignee_zip>
<consignee_tel><![CDATA[]]></consignee_tel>
<consignee_mobile><![CDATA[021-51086858]]></consignee_mobile>
<consignee_email><![CDATA[]]></consignee_email>
<sale_items>
  <bn><![CDATA[JCWL0002]]></bn>
  <name><![CDATA[基础物料名称二]]></name>
  <spec_name><![CDATA[WLGG0002]]></spec_name>
  <barcode><![CDATA[TMWL0002]]></barcode>
  <price>195.000</price>
  <nums>4</nums>
  <pmt_price>0.000</pmt_price>
  <sale_price>780.000</sale_price>
  <apportion_pmt>78.500</apportion_pmt>
  <sales_amount>701.500</sales_amount>
  <item_type><![CDATA[product]]></item_type>
  <type_name><![CDATA[普通]]></type_name>
  <sales_material_bn><![CDATA[XSWL0002]]></sales_material_bn>
  <sales_material_name><![CDATA[销售物料名称二]]></sales_material_name>
  <bn><![CDATA[jcw10001]]></bn>
  <name><![CDATA[基础物料名称一]]></name>
  <spec_name><![CDATA[]]></spec_name>
  <barcode><![CDATA[TMWL0001]]></barcode>
  <price>87.500</price>
  <nums>4</nums>
  <pmt_price>0</pmt_price>
  <sale_price>350.000</sale_price>
  <apportion_pmt>35.22</apportion_pmt>
  <sales_amount>350.000</sales_amount>
  <item_type><![CDATA[pkg]]></item_type>
  <type_name><![CDATA[促销]]></type_name>

```



```
<type_name><![CDATA[促销]]></type_name>
<sales_material_bn><![CDATA[WLPKG0001]]></sales_material_bn>
<sales_material_name><![CDATA[促销型销售物料名称一]]></sales_material_name>
<bn><![CDATA[JCWL0002]]></bn>
<name><![CDATA[基础物料名称二]]></name>
<spec_name><![CDATA[]]></spec_name>
<barcode><![CDATA[TMWL0002]]></barcode>
<price>75.000</price>
<nums>6</nums>
<pmt_price>0</pmt_price>
<sale_price>450.000</sale_price>
<apportion_pmt>45.28</apportion_pmt>
<sales_amount>450.000</sales_amount>
<item_type><![CDATA[pkg]]></item_type>
<type_name><![CDATA[促销]]></type_name>
<sales_material_bn><![CDATA[WLPKG0001]]></sales_material_bn>
<sales_material_name><![CDATA[促销型销售物料名称一]]></sales_material_name>
</sale_items>
</lists>
</response>
```

json数据, 例:

成功

```
{ "response": {
  "count": "6337",
  "lists": {
    "1355932064461265": {
      "shop_code": "shopex",
      "shop_name": "shopex",
      "order_no": "194276753441633",
      "member_name": "danny",
      "sale_no": "S20121219000001",
      "pay_method": "\u652f\u4ed8\u5b9d\u533b\u793a\u8303",
      "sale_time": "2012-12-08 18:50:30",
      "order_create_time": "2012-12-08 18:50:30",
      "pay_time": "2012-12-08 18:50:30",
      "ship_time": "2012-12-08 18:50:30",
      "order_check_op": "admin",
      "order_check_time": "2012-12-08 18:50:30",
      "goods_amount": "163.000",
      "freight_amount": "0.000",
      "additional_amount": "0.000",
      "has_tax": "\u5426",
      "pmt_amount": "0.000",
      "sale_amount": "163.000",
      "logi_name": "\u5706\u901a\u901f\u9012",
      "logi_no": "7104892994",
      "branch_name": "\u5c0f\u732a\u73ed\u7eb3",
      "delivery_no": "1211101100004",
      "consignee": "\u97e9\u7389\u5143",
      "consignee_area": "\u6e56\u5317-\u8346\u5dde\u5e02-\u6c99\u5e02\u533a",
      "consignee_addr": "\u6e56\u5317\u7701\u8346\u5dde\u5e02\u6c99\u5e02\u533a \u2026"
```

```
consignee_addr": "\u06e56\u05317\u07701 \u08346\u05dde\u05e02 \u06c99\u05e02\u0533a \u06e56\u05317\u07701\u08346\u05dde\u05e02\u06c99\u05e02\u0533a \u06e56\u05317\u07701\u08346\u05dde\u05e02\u06c99\u05e02\u0533a",
"consignee_zip": "200233",
"consignee_tel": "51086858",
"consignee_mobile": "",
"consignee_email": "",
"sale_items": {
  "1": {
    "bn": "JCWL0002",
    "name": "基础物料名称二",
    "spec_name": "WLG0002",
    "barcode": "TMWL0002",
    "price": "195.000",
    "nums": "4",
    "pmt_price": "0.000",
    "sale_price": "780.000",
    "apportion_pmt": "78.500",
    "sales_amount": "701.500",
    "item_type": "product",
    "type_name": "普通",
    "sales_material_bn": "XSWL0002",
    "sales_material_name": "销售物料名称二"
  },
  "2": {
    "bn": "jcw10001",
    "name": "基础物料名称一",
    "spec_name": "",
    "barcode": "TMWL0001",
    "price": "87.500",
    "nums": "4",
    "pmt_price": 0,
    "sale_price": "350.000",
    "apportion_pmt": 35.22,
    "sales_amount": "350.000",
    "item_type": "pkg",
    "type_name": "促销",
    "sales_material_bn": "WLPKG0001",
    "sales_material_name": "促销型销售物料名称一"
  },
  "3": {
    "bn": "JCWL0002",
    "name": "基础物料名称二",
    "spec_name": "",
    "barcode": "TMWL0002",
    "price": "75.000",
    "nums": "6",
    "pmt_price": 0,
    "sale_price": "450.000",
    "apportion_pmt": 45.28,
    "sales_amount": "450.000",
    "item_type": "pkg",
    "type_name": "促销",
    "sales_material_bn": "WLPKG0001",
    "sales_material_name": "促销型销售物料名称一"
  }
}
```

```
    },  
  }  
}  
}}
```

失败

```
{"error_response":{  
  "code":"e000001",  
  "msg":"system params lost or error",  
  "sub_msg":""  
}}
```

aftersales.getList 查询售后单信息(根据售后单创建时间)

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

批量获取一个时间段内的售后单信息数据

应用级参数

参数名称	是否必要	类型	描述
start_time	是	date	开始时间(售后单创建时间),例如2012-12-08 18:50:30
end_time	是	date	结束时间(同上)

返回结果

注意：已经做过特殊字符处理的字段如下：shop_code, shop_name, order_no, aftersale_no, return_change_no, member_name, member_mobile, refund_op,bn,name

XML数据，例：

成功

```
<response>
  <count>347</count>
  <lists>
    <shop_code><![CDATA[shopex]]></shop_code>
    <shop_name><![CDATA[shopex]]></shop_name>
    <order_no>174207669664609</order_no>
    <aftersale_no><![CDATA[A13543419237982]]></aftersale_no>
    <aftersale_apply_no>201212011415001401</aftersale_apply_no>
    <return_change_no>201212011413000634</return_change_no>
    <refund_apply_no></refund_apply_no>
    <aftersale_type><![CDATA[return]]></aftersale_type>
    <pay_method></pay_method>
    <refund_money>0.000</refund_money>
    <member_name><![CDATA[w668866]]></member_name>
    <member_mobile>13688568866</member_mobile>
    <check_op><![CDATA[admin]]></check_op>
    <quality_inspection_op><![CDATA[admin]]></quality_inspection_op>
    <refund_op><![CDATA[admin]]></refund_op>
    <apply_time>2012-12-08 18:50:30</apply_time>
    <check_time>2012-12-08 18:50:30</check_time>
    <quality_inspection_time>2012-12-08 18:50:30</quality_inspection_time>
    <refund_time>2012-12-08 18:50:30</refund_time>
    <aftersale_time>2012-12-08 18:50:30</aftersale_time>
    <aftersale_items>
      <bn>1114714954406150</bn>
      <name><![CDATA[test goods]]></name>
      <price>100.000</price>
      <nums>1</nums>
      <amount>100.000</amount>
      <branch_name><![CDATA[上海仓]]></branch_name>
      <apply_money>100.000</apply_money>
      <refund_money>100.000</refund_money>
    </aftersale_items>
  </lists>
</response>
```

json数据, 例:

成功

```
{
  "response": {
    "count": "347",
    "lists": {
      "2495": {
        "shop_code": "shopex",
        "shop_name": "shopex",
        "order_no": "174207669664609",
        "aftersale_no": "A13543419237982",
        "aftersale_apply_no": "201212011415001401",
        "return_change_no": "201212011413000634",
        "refund_apply_no": "201212011413000634",
        "aftersale_type": "return",
        "pay_method": null,
        "refund_money": "100.000",
        "member_name": "wx668866",
        "member_mobile": "13688568866",
        "check_op": "admin",
        "quality_inspection_op": "admin",
        "refund_op": "admin",
        "apply_time": "2012-12-08 18:50:30",
        "check_time": "2012-12-08 18:50:30",
        "quality_inspection_time": "2012-12-08 18:50:30",
        "refund_time": "2012-12-08 18:50:30",
        "aftersale_time": "2012-12-08 18:50:30",
        "aftersale_items": {
          "3272": {
            "bn": "1114714954406150",
            "name": "test goods",
            "price": "100.000",
            "nums": "1",
            "amount": "100.000",
            "branch_name": "\u5c0f\u732a\u73ed\u7eb3",
            "apply_money": "100.000",
            "refund_money": "100.000"
          }
        }
      }
    }
  }
}
```

iostock.getList 查询出入库明细

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

根据出入库时间来查询该时间段内的出入库明细

应用级参数

参数名称	是否必要	类型	描述
start_time	是	date	开始时间 2012-12-08 18:50:30
end_time	是	date	结束时间(同上)
branch_bn	否	string	仓库编号
page_no	否	number	页码, 默认1 第一页
page_size	否	number	每页数量, 最大100

返回结果

注意: 已经做过特殊字符处理的字段如下: iostock_bn, branch_bn, branch_name, bn, name, type, memo, original_bn

XML数据, 例:

成功

```
<response>
  <count>4</count>
  <iostock_price_num>19900.000</iostock_price_num>
  <unit_cost_num>0.000</unit_cost_num>
  <lists>
    <iostock_id>1441088465042587</iostock_id>
    <iostock_bn><![CDATA[I20150901000001]]></iostock_bn>
    <branch_bn><![CDATA[xck0101]]></branch_bn>
    <branch_name><![CDATA[新仓库测试]]></branch_name>
    <bn><![CDATA[h002]]></bn>
    <name><![CDATA[仙剑奇侠传6]]></name>
    <barcode><![CDATA[h002]]></barcode>
    <nums>100</nums>
    <type><![CDATA[采购入库]]></type>
    <iostock_time><![CDATA[2015-09-01 06:21:05]]></iostock_time>
    <memo><![CDATA[]]></memo>
    <original_bn><![CDATA[I20150901000001]]></original_bn>
    <iostock_price>50.000</iostock_price>
    <unit_cost>0.000</unit_cost>
    <iostock_id>1441088481252587</iostock_id>
    <iostock_bn><![CDATA[I20150901000002]]></iostock_bn>
    <branch_bn><![CDATA[xck0101]]></branch_bn>
    <branch_name><![CDATA[新仓库测试]]></branch_name>
    <bn><![CDATA[h003]]></bn>
    <name><![CDATA[仙剑玩偶]]></name>
    <barcode><![CDATA[h003]]></barcode>
    <nums>40</nums>
    <type><![CDATA[采购入库]]></type>
    <iostock_time><![CDATA[2015-09-01 06:21:21]]></iostock_time>
    <memo><![CDATA[]]></memo>
    <original_bn><![CDATA[I20150901000002]]></original_bn>
    <iostock_price>10.000</iostock_price>
    <unit_cost>0.000</unit_cost>
    <appropriation_no>S201512181125204</appropriation_no>
  </lists>
</response>
```

json数据，例：

成功


```
{
  "response": {
    "count": "2",
    "iostock_price_num": "19900.000",
    "unit_cost_num": "0.000",
    "lists": [
      {
        "iostock_id": "1441088465042587",
        "iostock_bn": "I20150901000001",
        "branch_bn": "xck0101",
        "branch_name": "新仓库测试",
        "bn": "h002",
        "name": "仙剑奇侠传6",
        "barcode": "h002",
        "nums": "100",
        "type": "采购入库",
        "iostock_time": "2015-09-01 06:21:05",
        "memo": "",
        "original_bn": "I20150901000001",
        "iostock_price": "50.000",
        "unit_cost": "0.000"
      },
      {
        "iostock_id": "1441088481252587",
        "iostock_bn": "I20150901000002",
        "branch_bn": "xck0101",
        "branch_name": "新仓库测试",
        "bn": "h003",
        "name": "仙剑玩偶",
        "barcode": "h003",
        "nums": "40",
        "type": "采购入库",
        "iostock_time": "2015-09-01 06:21:21",
        "memo": "",
        "original_bn": "I20150901000002",
        "iostock_price": "10.000",
        "unit_cost": "0.000",
        "appropriation_no": "S201512181125204"
      }
    ]
  }
}
```

po.add 新建采购单

- [详细描述](#)
- [应用级参数](#)
- [items json结构体描述](#)
- [返回结果](#)

详细描述

创建一个采购指令

应用级参数

参数名称	是否必要	类型	描述
name	是	string	采购单名称
vendor	是	string	供应商
po_type	是	string	采购方式 1 – 现购 2 – 赊购
branch_bn	是	string	到货仓库编号
delivery_cost	是	float	物流费用
deposit_balance	否	float	预付款
arrive_time	是	number	预计到货天数
memo	否	string	备注
operator	否	string	经办人
items	是	json	明细

items json结构体描述

参数名称	是否必要	类型	描述
bn	是	string	货号
name	是	string	货品名称
nums	是	string	数量
price	是	string	单价

返回结果

注意：已经做过特殊字符处理的字段如下：name, memo 货号和货品名称必须是仓库内已经存在的

XML数据，例：

成功

```
<response>
  <rsp>succ</rsp>
  <data>201305201617000180</data>
</response>
```

失败

```
<response>
  <rsp>fail</rsp>
  <msg>系统异常</msg>
</response>
```

json数据，例：

成功

```
{"response":{
  "rsp":"succ",
  "data":" 201305201617000180" //此为成功后的采购单号
}}
```

失败

```
{"response":{
  "rsp":"fail",
  "msg":"系统异常"
}}
```

transfer.add 创建出入单

- [详细描述](#)
- [应用级参数](#)
- [items json结构体描述](#)
- [返回结果](#)

详细描述

创建一个直接出入库的指令

应用级参数

参数名称	是否必要	类型	描述
name	是	string	入库单名称
vendor	否	string	供应商
branch_bn	是	string	仓库编号
delivery_cost	否	float	出入库费用
t_type	是	string	出入库类型 E – 直接入库 A – 直接出库 G – 赠品入库 F – 赠品出库 K – 样品入库 J – 样品出库 Y – 分销入库 Z – 分销出库
memo	否	string	备注
operator	否	string	经办人
confirm	否	string	是否已审核（ Y - 是 N - 否 ） 默认值： N
items	是	json	明细

items json结构体描述

参数名称	是否必要	类型	描述
bn	是	string	货号
name	是	string	货品名称
nums	是	string	数量
price	是	string	单价

返回结果

注意： 已经做过特殊字符处理的字段如下： name, vendor, type, branch_bn, delivery_cost, operator, memo, confirm, ['items']['bn'], ['items']['name'], ['items']['nums'], ['items']['price']

XML数据，例：

成功

```
<response>
  <rsp>![CDATA[succ]]</rsp>
  <data>![CDATA[A201305210001]]</data>
</response>
```

失败

```
<response>
  <rsp>fail</rsp>
  <msg>系统异常</msg>
</response>
```

json数据，例：

成功

```
{"response":{
  "rsp":"succ",
  "data":"A201305210001" //此为成功后的出入库单号
}}
```

失败

```
{"response":{
  "rsp":"fail",
  "msg":"系统异常"
}}
```

pkg.getList 查询捆绑商品信息

- [详细描述](#)
- [应用级参数](#)
- [items json结构体描述](#)
- [返回结果](#)

详细描述

获取捆绑商品信息

应用级参数

参数名称	是否必要	类型	示例	描述
page_no	否	number	1	页码,默认:1
page_size	否	number	40	每页最大数量,最大值: 100, 默认: 40

items json结构体描述

参数名称	是否必要	类型	描述
pkg_bn	是	string	捆绑货号
pkg_name	是	string	捆绑名称
weight	是	number	重量
Products	是	product[]	货品结构

返回结果

注意：已经做过特殊字符处理的字段如下：pkg_bn, pkg_name, bn, name

XML数据，例：

成功

```
<response>
  <count>2</count>
  <lists>
    <pkg_bn><![CDATA[xj005]]></pkg_bn>
    <pkg_name><![CDATA[仙剑模型]]></pkg_name>
    <weight>50.000</weight>
    <products>
      <bn><![CDATA[h002]]></bn>
      <name><![CDATA[仙剑奇侠传6]]></name>
      <nums>20</nums>
      <price>88.000</price>
      <weight>56.000</weight>
    </products>
    <pkg_bn><![CDATA[abc001]]></pkg_bn>
    <pkg_name><![CDATA[abc捆绑名称]]></pkg_name>
    <weight>20.000</weight>
    <products>
      <bn><![CDATA[h002]]></bn>
      <name><![CDATA[仙剑奇侠传6]]></name>
      <nums>3</nums>
      <price>88.000</price>
      <weight>56.000</weight>
    </products>
  </lists>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "count": 2,
    "lists": {
      "1": {
        "pkg_bn": "xj005",
        "pkg_name": "仙剑模型",
        "weight": "50.000",
        "products": {
          "2": {
            "bn": "h002",
            "name": "仙剑奇侠传6",
            "nums": "20",
            "price": "88.000",
            "weight": "56.000"
          }
        }
      },
      "2": {
        "pkg_bn": "abc001",
        "pkg_name": "abc捆绑名称",
        "weight": "20.000",
        "products": {
          "2": {
            "bn": "h002",
            "name": "仙剑奇侠传6",
            "nums": "3",
            "price": "88.000",
            "weight": "56.000"
          }
        }
      }
    }
  }
}
```


stock.getAll 商品库存查询接口

- [详细描述](#)
- [应用级参数](#)
- [items json结构体描述](#)
- [返回结果](#)

详细描述

返回商品库存信息

应用级参数

参数名称	是否必要	类型	示例	描述
goods_bn	否	string		商品编码
brand_name	否	string		品牌
type_name	否	string		类型
page_no	否	number	1	页码
page_size	否	number	100	每页数量

返回结果

注意：已经做过特殊字符处理的字段如下：bn,name, branch_bn, branch_name

XML数据，例：

成功

```
<response>
  <lists>
    <bn>
      <![CDATA[h002]]>
    </bn>
    <name>
      <![CDATA[仙剑奇侠传6]]>
    </name>
    <store>150</store>
    <store_freeze>6</store_freeze>
    <spec_info>
      <![CDATA[]]>
    </spec_info>
    <branchstore>
      <branch_bn>
        <![CDATA[csck001]]>
      </branch_bn>
      <branch_name>
        <![CDATA[测试仓库111]]>
      </branch_name>
      <store>50</store>
      <store_freeze>0</store_freeze>
      <arrive_store>0</arrive_store>
      <branch_bn>
        <![CDATA[xck0101]]>
      </branch_bn>
      <branch_name>
        <![CDATA[新仓库测试]]>
      </branch_name>
      <store>100</store>
      <store_freeze>0</store_freeze>
      <arrive_store>0</arrive_store>
    </branchstore>
  </lists>
  <count>1</count>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "lists": {
      "1": {
        "bn": "678",
        "name": "nike运动鞋",
        "store": "0",
        "store_freeze": "0",
        "spec_info": "42码",
        "branchstore": [ ]
      },
      "2": {
        "bn": "h002",
        "name": "仙剑奇侠传6",
        "store": "150",
        "store_freeze": "6",
        "spec_info": "",
        "branchstore": {
          "1": {
            "branch_bn": "csck001",
            "branch_name": "测试仓库111",
            "store": "50",
            "store_freeze": "0",
            "arrive_store": "0"
          },
          "2": {
            "branch_bn": "xck0101",
            "branch_name": "新仓库测试",
            "store": "100",
            "store_freeze": "0",
            "arrive_store": "0"
          }
        }
      }
    },
    "count": 2
  }
}
```

stock.getDetailList 仓库库存查询接口

- [详细描述](#)
- [应用级参数](#)
- [items json结构体描述](#)
- [返回结果](#)

详细描述

返回各仓库中商品库存

应用级参数

参数名称	是否必要	类型	示例	描述
product_bn	否	string		货号
branch_bn	否	string		仓库编码
page_no	否	number	1	页码
page_size	否	number	100	每页数量

返回结果

注意：已经做过特殊字符处理的字段如下：product_bn, product_name, product_spec, branch_bn, branch_name

XML数据，例：

成功

```
<response>
  <lists>
    <store>50</store>
    <store_freeze>0</store_freeze>
    <store_in_transit>0</store_in_transit>
    <product_bn>
      <![CDATA[h002]]>
    </product_bn>
    <product_name>
      <![CDATA[仙剑奇侠传6]]>
    </product_name>
    <product_spec>
      <![CDATA[]]>
    </product_spec>
    <branch_bn>
      <![CDATA[cscck001]]>
    </branch_bn>
    <branch_name>
      <![CDATA[测试仓库111]]>
    </branch_name>
    <store>100</store>
    <store_freeze>0</store_freeze>
    <store_in_transit>0</store_in_transit>
    <product_bn>
      <![CDATA[h002]]>
    </product_bn>
    <product_name>
      <![CDATA[仙剑奇侠传6]]>
    </product_name>
    <product_spec>
      <![CDATA[]]>
    </product_spec>
    <branch_bn>
      <![CDATA[xcck0101]]>
    </branch_bn>
    <branch_name>
      <![CDATA[新仓库测试]]>
    </branch_name>
  </lists>
  <count>2</count>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "lists": {
      "1": {
        "store": "50",
        "store_freeze": "0",
        "store_in_transit": "0",
        "product_bn": "h002",
        "product_name": "仙剑奇侠传6",
        "product_spec": "",
        "branch_bn": "csck001",
        "branch_name": "测试仓库111"
      },
      "2": {
        "store": "100",
        "store_freeze": "0",
        "store_in_transit": "0",
        "product_bn": "h002",
        "product_name": "仙剑奇侠传6",
        "product_spec": "",
        "branch_bn": "xck0101",
        "branch_name": "新仓库测试"
      },
    },
    "count": 2
  }
}
```

po.getList 采购单查询接口

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

返回采购单信息

应用级参数

参数名称	是否必要	类型	示例	描述
po_bn	否	string		采购单编号
supplier	否	string		供应商名称
start_time	否	string		开始时间
end_time	否	string		结束时间
eo_status	否	string	1	入库状态
check_status	否	string	1	审核状态
po_status	否	string	1	采购状态
statement_status	否	string	1	结算状态
page_no	否	number	1	页码
page_size	否	number	100	每页数量

返回结果

注意：已经做过特殊字符处理的字段如下：po_bn, product_bn, product_name

XML数据，例：

成功

```
<response>
  <po_id>1</po_id>
  <po_name><![CDATA[20150831采购单]]></po_name>
  <po_bn><![CDATA[I201508311517009516]]></po_bn>
  <supplier>1</supplier>
  <po_time><![CDATA[2015-08-31 07:18:47]]></po_time>
  <amount>2500.000</amount>
  <operator><![CDATA[admin]]></operator>
  <branch>2</branch>
  <po_status>1</po_status>
  <statement_status>1</statement_status>
  <check_status>1</check_status>
  <eo_status>0</eo_status>
  <logistic_fee/>
  <item_cost>2500.000</item_cost>
  <deposit>0.000</deposit>
  <deposit_balance>0.000</deposit_balance>
  <items>
    <product_bn><![CDATA[h002]]></product_bn>
    <product_name><![CDATA[仙剑奇侠传6]]></product_name>
    <price>50.000</price>
    <num>50</num>
    <in_num>0</in_num>
    <status>1</status>
  </items>
</response>
```

json数据，例：

成功


```
{
  "response": [
    {
      "po_id": "1",
      "po_name": "20150831采购单",
      "po_bn": "I201508311517009516",
      "supplier": "1",
      "po_time": "2015-08-31 07:18:47",
      "amount": "2500.000",
      "operator": "admin",
      "branch": "2",
      "po_status": "1",
      "statement_status": "1",
      "check_status": "1",
      "eo_status": "0",
      "logistic_fee": null,
      "item_cost": "2500.000",
      "deposit": "0.000",
      "deposit_balance": "0.000",
      "items": [
        {
          "product_bn": "h002",
          "product_name": "仙剑奇侠传6",
          "price": "50.000",
          "num": "50",
          "in_num": "0",
          "status": "1"
        }
      ]
    }
  ]
}
```

invoice.getList 发票列表

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

获取订单发票的信息列表

应用级参数

参数名称	是否必要	类型	描述
mode	否	string	开票方式 1: 电子发票 ; 0: 纸质发票 ; 不填不区分
is_status	否	string	开票状态 1: 已开票 ; 0: 未发票 ; 2: 已作废 ; 不填不区分
start_time	是	date	开始时间(发票创建时间) 例如2012-12-08 18:50:30
end_time	是	date	结束时间(发票创建时间) 例如2012-12-08 18:50:30
page_no	否	number	页码
page_size	否	number	每页条数(最大100条)

返回结果

注意：已经做过特殊字符处理的字段如下：['list']['product']['name'], ['list']['product']['order_items']['name']

XML数据，例：

成功

```
<response>
  <list>
    <id>16</id>
    <order_id>22</order_id>
    <order_bn>201608011510005294</order_bn>
    <type_id>0</type_id>
    <mode><![CDATA[纸质发票]]></mode>
    <amount>12.000</amount>
    <cost_tax>2.550</cost_tax>
    <tax_rate>17</tax_rate>
    <title><![CDATA[haha01]]></title>
    <is_status><![CDATA[已开票]]></is_status>
    <sync>0</sync>
    <is_print>1</is_print>
    <invoice_code/>
    <invoice_no><![CDATA[fapiaohao01]]></invoice_no>
    <remarks><![CDATA[备注内容]]></remarks>
```

```

<operator>1</operator>
<print_num>1</print_num>
<create_time>1470036810</create_time>
<dateline>1470037135</dateline>
<update_time>1470037135</update_time>
<tax_company><![CDATA[王俊]]></tax_company>
<ship_area><![CDATA[上海 上海市 普陀区]]></ship_area>
<ship_addr><![CDATA[上海 上海市 普陀区曹杨路]]></ship_addr>
<ship_tel>13918458888</ship_tel>
<ship_tax><![CDATA[]]></ship_tax>
<ship_bank><![CDATA[]]></ship_bank>
<ship_bank_no><![CDATA[]]></ship_bank_no>
<payee_name><![CDATA[电子 发票开发]]></payee_name>
<tax_no>440300568519737</tax_no>
<address><![CDATA[曹杨路]]></address>
<telephone>13918458888</telephone>
<payee_operator><![CDATA[王俊]]></payee_operator>
<bank><![CDATA[建行]]></bank>
<bank_no>20020</bank_no>
<content><![CDATA[礼品]]></content>
<product>
  <obj_id>26</obj_id>
  <obj_alias><![CDATA[goods]]></obj_alias>
  <name><![CDATA[销售物料01]]></name>
  <price>10.000</price>
  <pmt_price>0.000</pmt_price>
  <sale_price>10.000</sale_price>
  <quantity>1</quantity>
  <order_items>
    <item_id>24</item_id>
    <name><![CDATA[基础物料01]]></name>
    <price>10.000</price>
    <pmt_price>0.000</pmt_price>
    <sale_price>10.000</sale_price>
    <nums>1</nums>
    <addon>
      <product_attr/>
    </addon>
  </order_items>
</product>
<shop_name><![CDATA[淘嗨]]></shop_name>
<shop_id><![CDATA[c9806ea45153df1fead3fc91d7dc4a68]]></shop_id>
</list>
</response>

```

json数据，例：

成功

```

{
  "response": {
    "list": [
      {

```

```
"id": "16",
"order_id": "22",
"order_bn": "201608011510005294",
"type_id": "0",
"mode": "纸质发票",
"amount": "12.000",
"cost_tax": "2.550",
"tax_rate": "17",
"title": "haha01",
"is_status": "已开票",
"sync": "0",
"is_print": "1",
"invoice_code": null,
"invoice_no": "fapiaohao01",
"remarks": "备注内容",
"operator": "1",
"print_num": "1",
"create_time": "1470036810",
"dateline": "1470037135",
"update_time": "1470037135",
"tax_company": "王俊",
"ship_area": "上海 上海市 普陀区",
"ship_addr": "上海 上海市 普陀区曹杨路",
"ship_tel": "13918458888",
"ship_tax": "",
"ship_bank": "",
"ship_bank_no": "",
"payee_name": "电子 发票开发",
"tax_no": "440300568519737",
"address": "曹杨路",
"telephone": "13918458888",
"payee_operator": "王俊",
"bank": "建行",
"bank_no": "20020",
"content": "礼品",
"product": [
  {
    "obj_id": "26",
    "obj_alias": "goods",
    "name": "销售物料01",
    "price": "10.000",
    "pmt_price": "0.000",
    "sale_price": "10.000",
    "quantity": "1",
    "order_items": [
      {
        "item_id": "24",
        "name": "基础物料01",
        "price": "10.000",
        "pmt_price": "0.000",
        "sale_price": "10.000",
        "nums": "1",
        "addon": {
```

```
        "product_attr": null
      }
    }
  ]
},
"shop_name": "淘嗨",
"shop_id": "c9806ea45153df1fead3fc91d7dc4a68"
}
]
}
}
```

invoice.update 更新纸质发票的打印信息

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

更新纸质发票的打印信息

应用级参数

参数名称	是否必要	类型	描述
order_bn	是	string	订单号
shop_id	是	string	前端店铺ID
invoice_no	是	string	发票号

返回结果

注意：失败提示包含：请填写必填参数，此订单发票信息不存在，必须为纸质发票，此发票已作废。

XML数据，例：

成功

```
<response><msg><![CDATA[success]]></msg><message><![CDATA[更新发票的打印状态完成.]]></message></response>
```

json数据，例：

成功

```
{"response":{"msg":"success","message":"更新发票的打印状态完成."}}
```

goods.getList 批量获取商品信息

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

批量获取商品信息

应用级参数

参数名称	是否必要	类型	描述
brand_name	否	string	商品品牌名称
type_name	否	string	商品类型名称
goods_bn	否	string	商品编码
start_lastmodify	否	string	修改开始时间,格式:2014-05-26 16:18:01
end_lastmodify	否	string	修改结束时间,格式:2014-05-26 16:18:02
page_no	是	number	页码
page_size	是	string	每页条数(最大100条)

返回结果

注意：已经做过特殊字符处理的字段如下： goods_bn, goods_name, goods_type, goods_brand, unit, product_bn, spec_info, barcode

XML数据，例：

成功

```
<response>
  <list>
    <goods_bn><![CDATA[wo001]]></goods_bn>
    <goods_name><![CDATA[仙剑玩偶]]></goods_name>
    <goods_type><![CDATA[电脑单机游戏]]></goods_type>
    <goods_brand><![CDATA[仙剑奇侠传6]]></goods_brand>
    <unit><![CDATA[个]]></unit>
    <lastmodify><![CDATA[2015-08-25 08:35:45]]></lastmodify>
    <products>
      <product_bn><![CDATA[h003]]></product_bn>
      <spec_info><![CDATA[]]></spec_info>
      <store>0</store>
      <store_freeze>0</store_freeze>
      <weight>2.000</weight>
      <cost>10.000</cost>
      <price>5.000</price>
      <mktprice>0.000</mktprice>
      <barcode><![CDATA[h003]]></barcode>
    </products>
  </list>
  <count>1</count>
</response>
```

json数据，例：

成功


```
{
  "response": {
    "list": [
      {
        "goods_bn": "wo001",
        "goods_name": "仙剑玩偶",
        "goods_type": "电脑单机游戏",
        "goods_brand": "仙剑奇侠传6",
        "unit": "个",
        "lastmodify": "2015-08-25 08:35:45",
        "products": [
          {
            "product_bn": "h003",
            "spec_info": "",
            "store": "0",
            "store_freeze": "0",
            "weight": "2.000",
            "cost": "10.000",
            "price": "5.000",
            "mktprice": "0.000",
            "barcode": "h003"
          }
        ]
      }
    ],
    "count": 1
  }
}
```

delivery.getList 批量获取发货单信息

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

批量获取发货单信息

应用级参数

参数名称	是否必要	类型	描述
branch_name	否	string	仓库
shop_name	否	string	来源店铺
receive_area	否	string	收货地区
corp_name	否	string	快递公司
create_starttime	是	string	创建时间(开始)格式:2014-05-26 16:18:01
create_endtime	否	string	创建时间(结束)
ship_starttime	否	string	发货时间(开始)格式:2014-05-26 16:18:01
ship_endtime	否	string	发货时间(结束)格式:2014-05-26 16:18:02

返回结果

注意：已经做过特殊字符处理的字段如下：shop_code, shop_name, order_no, member_name, delivery_bn, freight_amount, logi_name, logi_no, branch_name, branch_bn, delivery_no, consignee, consignee_addr, consignee_zip, consignee_mobile, consignee_email, mark_memo, custom_memo, bn, product_name

XML数据，例：

成功

```
<response>
  <count>1</count>
  <lists>
    <shop_code><![CDATA[wanpy14358]]></shop_code>
    <shop_name><![CDATA[当当网顽皮旗舰店]]></shop_name>
    <order_no>2014072030071857</order_no>
    <member_name></member_name>
    <delivery_bn>1407171100001</delivery_bn>
    <create_time><![CDATA[2014-07-17 14:19:46]]></create_time>
    <ship_time><![CDATA[]]></ship_time>
    <freight_amount></freight_amount>
    <logi_name><![CDATA[宅急送]]></logi_name>
    <logi_no>55</logi_no>
    <branch_name><![CDATA[上海仓库]]></branch_name>
    <branch_bn>3</branch_bn>
    <delivery_no>1407171100001</delivery_no>
    <consignee><![CDATA[孙玥]]></consignee>
    <consignee_area><![CDATA[广东-深圳市-福田区]]></consignee_area>
    <consignee_addr><![CDATA[]]></consignee_addr>
    <consignee_zip>518000</consignee_zip>
    <consignee_tel><![CDATA[]]></consignee_tel>
    <consignee_mobile>18676703649</consignee_mobile>
    <consignee_email><![CDATA[]]></consignee_email>
    <items>
      <bn>6927749802662</bn>
      <product_name><![CDATA[顽皮成犬狗狗鲜封包 妙鲜包 狗粮伴侣 鸡肉奶酪鲜封包80g*15袋]]><
/product_name>
      <price>9</price>
      <nums>1</nums>
    </items>
  </lists>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "count": "1",
    "lists": {
      "1": {
        "shop_code": "wanpy14358",
        "shop_name": "当当网顽皮旗舰店",
        "order_no": "2014072030071857",
        "member_name": null,
        "delivery_bn": "1407171100001",
        "create_time": "2014-07-17 14:19:46",
        "ship_time": "",
        "freight_amount": null,
        "logi_name": "宅急送",
        "logi_no": "55",
        "branch_name": "上海仓库",
        "branch_bn": "3",
        "delivery_no": "1407171100001",
        "consignee": "孙玥",
        "consignee_area": "广东-深圳市-福田区",
        "consignee_addr": "",
        "consignee_zip": "518000",
        "consignee_tel": "",
        "consignee_mobile": "18676703649",
        "consignee_email": "",
        "items": {
          "1": {
            "bn": "6927749802662",
            "product_name": "顽皮成犬狗狗鲜封包 妙鲜包 狗粮伴侣 鸡肉奶酪鲜封包80g*15
袋",
            "price": 9,
            "nums": "1"
          }
        }
      }
    }
  }
}
```

goods.add 添加商品接口

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

添加商品的方法

应用级参数

参数名称	是否必要	类型	描述
brand_name	否	string	品牌名称
type_name	否	string	商品类型
goods_name	是	string	商品名称
goods_bn	否	string	商品编号
unit	否	string	计量单位
is_serial	否	string	唯一码开启
sale_price	否	string	销售价
cost_price	否	string	成本价
product_bn	是	string	货号
barcode	是	string	条形码
weight	否	string	重量

返回结果

注意：已经做过特殊字符处理的字段如下：bn,name,type_id,brand_id,barcode,['product']['price']
['price']['price'], ['product']['price']['cost']['price'],['product']['bn'],['product']['barcode'], ['product']
['weight'],['product']['unit']

XML数据，例：

成功

```
<response>
  <rsp><![CDATA[succ]]></rsp>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "rsp": "succ"
  }
}
```

sales.getSalesAmount 获取销售单统计数据

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

获取一段时间段内的销售单总额数据

应用级参数

参数名称	是否必要	类型	描述
start_time	是	string	开始时间
end_time	是	string	结束时间
page_no	是	string	页码
page_size	是	string	每页数量
shop_bn	否	string	店铺编码 多个店铺编码之间，用#分隔

返回结果

XML数据，例：

成功

```
<response>
  <lists>
    <sales_amount>50137.120</sales_amount>
    <cost_freight>1000.000</cost_freight>
    <discount>1016.100</discount>
    <additional_costs>1.100</additional_costs>
    <shop_bn><![CDATA[SD123]]></shop_bn>
    <sales_amount>63.000</sales_amount>
    <cost_freight>10.000</cost_freight>
    <discount>0.000</discount>
    <additional_costs>1.000</additional_costs>
    <shop_bn><![CDATA[蘑菇街]]></shop_bn>
    <sales_amount>10.000</sales_amount>
    <cost_freight>0.000</cost_freight>
    <discount>0.000</discount>
    <additional_costs>0.000</additional_costs>
    <shop_bn><![CDATA[yunzhan]]></shop_bn>
  </lists>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "lists": [
      {
        "sales_amount": "50137.120",
        "cost_freight": "1000.000",
        "discount": "1016.100",
        "additional_costs": "1.100",
        "shop_bn": "SD123"
      },
      {
        "sales_amount": "63.000",
        "cost_freight": "10.000",
        "discount": "0.000",
        "additional_costs": "1.000",
        "shop_bn": "蘑菇街"
      },
      {
        "sales_amount": "10.000",
        "cost_freight": "0.000",
        "discount": "0.000",
        "additional_costs": "0.000",
        "shop_bn": "yunzhan"
      }
    ]
  }
}
```


purchasereturn.getList 获取采购退货单数据

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

获取指定条件下的采购退货单列表

应用级参数

参数名称	是否必要	类型	描述
rp_bn	否	string	退货单编号
supplier	否	string	供应商名称
returned_time	否	string	退货日期
branch	否	string	仓库
return_status	否	string	退货单状态1 默认2 退货完成 3 出库拒绝
page_no	是	string	页码
page_size	是	string	每页数量

返回结果

注意：已经做过特殊字符处理的字段如下：product_bn, product_name, supplier, branch

XML数据，例：

成功

```
<response>
  <po_name>
    <![CDATA[退货单测试01]]>
  </po_name>
  <rp_bn>
    <![CDATA[H201509010817004536]]>
  </rp_bn>
  <supplier>
    <![CDATA[g01-s]]>
  </supplier>
  <po_time>1441094637</po_time>
  <amount>20.000</amount>
  <operator>
    <![CDATA[admin]]>
  </operator>
  <branch>
    <![CDATA[新仓库测试]]>
  </branch>
  <return_status>1</return_status>
  <logistic_fee>0.000</logistic_fee>
  <item_cost>20.000</item_cost>
  <items>
    <product_bn>
      <![CDATA[h002]]>
    </product_bn>
    <product_name>
      <![CDATA[仙剑奇侠传6]]>
    </product_name>
    <num>2</num>
  </items>
</response>
```

json数据，例：

成功

```
{
  "response": [
    {
      "po_name": "退货单测试01",
      "rp_bn": "H201509010817004536",
      "supplier": "g01-s",
      "po_time": "1441094637",
      "amount": "20.000",
      "operator": "admin",
      "branch": "新仓库测试",
      "return_status": "1",
      "logistic_fee": "0.000",
      "item_cost": "20.000",
      "items": {
        "product_bn": "h002",
        "product_name": "仙剑奇侠传6",
        "num": "2"
      }
    }
  ]
}
```

purchasereturn.add 新建采购退货单

- [详细描述](#)
- [应用级参数](#)
- [items json结构体描述](#)
- [返回结果](#)

详细描述

创建采购退货单

应用级参数

参数名称	是否必要	类型	描述
name	是	string	退货单名称
supplier_bn	是	string	供应商编码
branch_bn	是	string	退货仓库编码
operator	是	string	经办人
delivery_cost	否	string	物流费用
logi_no	否	string	物流单号
emergency	否	string	是否为特别退货单（是/否，默认为否）
memo	否	string	备注
items	是	Json_encode	明细

items json结构体描述

参数名称	是否必要	类型	描述
bn	是	string	货号
price	是	number	价格
nums	是	number	采购退货商品的数量

返回结果

XML数据，例：

成功

```
<response>
  <rsp><![CDATA[succ]]></rsp>
</response>
```

失败

```
<response>
  <rsp><![fail]> </rsp >
  <msg><![货品货号不存在!]> </msg>
</response>
```

json数据，例：

成功

```
{"response":{"rsp":"succ"}}
```

失败

```
{
  "response":{
    "rsp":"fail","msg":"货品货号不存在!"
  }
}
```

goods.edit 商品编辑

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

编辑现有商品的信息

应用级参数

参数名称	是否必要	类型	描述
brand_name	否	string	品牌名称
type_name	否	string	商品类型
goods_name	否	string	商品名称
goods_bn	否	string	商品编码
unit	否	string	计量单位
is_serial	否	string	唯一码是否开启（是/否）
sale_price	否	number	销售价
cost_price	否	number	成本价
product_bn	是	string	货号
barcode	否	string	条形码
weight	否	number	重量（克）

返回结果

XML数据，例：

成功

```
<response>
  <rsp><![succ]> </rsp >
</response>
```

失败

```
<response>
  <rsp><![fail]> </rsp >
  <msg><![货品货号不存在!]> </msg>
</response>
```

json数据，例：

成功

```
{"response":{"rsp":"succ"}}
```

失败

```
{  
  "response":{  
    "rsp":"fail","msg":"货品货号不存在!"  
  }  
}
```

appropriation.add 新建调拨单

- [详细描述](#)
- [应用级参数](#)
- [items json结构体描述](#)
- [返回结果](#)

详细描述

创建调拨单

应用级参数

参数名称	是否必要	类型	描述
appropriation_type	是	number	调拨单类型（1：直接调拨 / 2：出入库调拨）
from_branch	是	string	调出仓库编号
to_branch	是	string	调入仓库编号
is_check（协同版参数）	否	string	调拨出库单是否自动审核（是/否，默认为否，出入库调拨类型时可用）
logi_name	否	string	物流公司名称
operator	否	string	经办人
memo	否	string	备注
items	是	json_encode	明细

items json结构体描述

参数名称	是否必要	类型	描述
bn	是	string	货号
nums	是	number	调拨数量

返回结果

XML数据，例：

成功

```
<response>
  <rsp><![CDATA[succ]]></rsp>
  <msg><![CDATA[调拨单创建成功]]></msg>
</response>
```

失败


```
<response>
  <rsp><![fail]> </rsp >
  <msg><![在测试仓库不存在货号test!]> </msg>
</response>
```

json数据，例：

成功

```
{
  "response":
    {
      "rsp": "succ",
      "msg": "调拨单创建成功"
    }
}
```

失败

```
{
  "response": {
    "rsp": "fail", "msg": "在测试仓库不存在货号test!"
  }
}
```

basicmaterial.add添加基础物料

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

添加基础物料

应用级参数

参数名称	是否必须	类型	描述
material_name	是	string	基础物料名称
material_bn	是	String	基础物料编号
material_type	是	Number	基础物料类型(1代表普通类型、2代表组合类型)
serial_number	否	String	是否启用唯一码(true 开启、false 关闭)
visibled	是	Number	是否可售(1代表可售、2代表停售)
materil_code	是	String	条码
unit	否	String	计量单位
retail_price	否	String	销售价
cost	否	String	成本价
weight	否	String	重量
bind_info	否	String	组合类物料信息(test001:1 test002:2)
use_expire	否	Number	是否开启保质期监控(1代表开启、2代表关闭)
warn_day	否	Number	预警天数配置
quit_day	否	Number	自动退出库存天数配置

返回结果

XML数据，例：

失败样式：

```
<response>
  <rsp><![fail]> </rsp >
  <msg><![当前新增的物料编码已被使用，不能重复]> </msg>
</response>
```

成功样式：

```
<response>
  <rsp><![succ]> </rsp >
</response>
```

json数据，例：

失败样式：

```
{
  "response":
    {
      "rsp":"fail","msg":"当前新增的物料编码已被使用，不能重复"
    }
}
```

成功样式：

```
{"response":{"rsp":"succ"}}
```

basicmaterial.edit编辑基础物料

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

编辑基础物料

应用级参数

参数名称	是否必须	类型	描述
material_name	是	String	基础物料名称
material_bn	是	String	基础物料编号
material_type	是	Number	基础物料类型(1代表普通类型、2代表组合类型)
serial_number	否	String	是否启用唯一码(true 开启、false 关闭)
visibled	是	Number	是否可售(1代表可售、2代表停售)
unit	否	String	计量单位
retail_price	否	String	销售价
cost	否	String	成本价
weight	否	String	重量
bind_info	否	String	组合类物料信息(test001:1 test002:2)
use_expire	否	Number	是否开启保质期监控(1代表开启、2代表关闭)
warn_day	否	Number	预警天数配置
quit_day	否	Number	自动退出库存天数配置

返回结果

XML数据，例：

失败样式：

```
<response>
  <rsp><![fail]> </rsp >
  <msg><![当前物料不存在]> </msg>
</response>
```

成功样式：

```
<response>
  <rsp><![succ]> </rsp >
</response>
```

json数据，例：

失败样式：

```
{
  "response":
    {
      "rsp":"fail","msg":"当前物料不存在"
    }
}
```

成功样式：

```
{"response":{"rsp":"succ"}}
```

basicmaterial.getList获取基础物料

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

获取指定条件下的基础物料列表

应用级参数

参数名称	是否必须	类型	描述
material_bn	否	String	基础物料编号
page_no	是	String	页码
page_size	是	String	每页数量

返回结果

XML数据，例：

```
<response>
  <lists>
    <material_name><![CDATA[b002]]></material_name>
    <material_bn><![CDATA[b002]]></material_bn>
    <visible>1</visible>
    <serial_number><![CDATA[false]]></serial_number>
    <type>1</type>
    <cost>50.000</cost>
    <retail_price>100.000</retail_price>
    <weight>5.000</weight>
    <unit><![CDATA[包]]></unit>
    <barcode><![CDATA[b002]]></barcode>
    <material_name><![CDATA[b001]]></material_name>
  </lists>
  <count>76</count>
</response>
```

json数据，例：

失败样式：

```
{
  response:{
    "lists":{
      "92":{"material_name":"b002","material_bn":"b002","visibled":"1","serial_number":
"false","type":"1","cost":"50.000","retail_price":"100.000","weight":"5.000","unit":"包","
barcode":"b002"}
      "count":76
    }
  }
}
```

salesmaterial.add添加销售物料

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

添加销售物料

应用级参数

参数名称	是否必须	类型	描述
sales_material_name	是	String	销售物料名称
sales_material_bn	是	String	销售物料编号
sales_material_type	是	Number	销售物料类型(1代表普通类型、2代表促销类型、3代表赠品类型)
unit	否	String	计量单位
retail_price	否	String	销售价
bind_info	否	String	组合类物料信息(test001:1x100 test002:2x200)

返回结果

XML数据，例：

失败样式：

```
<response>
  <rsp><![fail]> </rsp >
  <msg><![当前新增的物料编码已被使用，不能重复]> </msg>
</response>
```

成功样式：

```
<response>
  <rsp><![succ]> </rsp >
</response>
```

json数据，例：

失败样式：


```
{
  "response":
  {
    "rsp":"fail","msg":"当前新增的物料编码已被使用，不能重复"
  }
}
```

成功样式：

```
{"response":{"rsp":"succ"}}
```

salesmaterial.edit编辑销售物料

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

编辑销售物料

应用级参数

参数名称	是否必须	类型	描述
sales_material_name	是	String	销售物料名称
sales_material_bn	是	String	销售物料编号
sales_material_type	是	Number	销售物料类型(1代表普通类型、2代表促销类型、3代表赠品类型)
unit	否	String	计量单位
retail_price	否	String	销售价
bind_info	否	String	组合类物料信息(test001:1x100 test002:2x200)

返回结果

XML数据，例：

失败样式：

```
<response>
  <rsp><![fail]> </rsp >
  <msg><![当前物料不存在]> </msg>
</response>
```

成功样式：

```
<response>
  <rsp><![succ]> </rsp >
</response>
```

json数据，例：

失败样式：

```
{
  "response":
  {
    "rsp":"fail","msg":"当前物料不存在"
  }
}
```

成功样式:

```
{"response":{"rsp":"succ"}}
```

salesmaterial.getList获取销售物料

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

获取指定条件下的销售物料列表

应用级参数

参数名称	是否必须	类型	描述
sales_material_bn	否	String	销售物料编号
page_no	是	String	页码
page_size	是	String	每页数量

返回结果

XML数据，例：

```
<response>
  <lists>
    <sales_material_bn><![CDATA[testC003]]></sales_material_bn>
    <sales_material_name><![CDATA[testC003]]></sales_material_name>
    <unit><![CDATA[袋]]></unit>
    <sales_material_type>1</sales_material_type>
    <retail_price>12.000</retail_price>
    <shop_id><![CDATA[c0b540134794156ebe6ed58cedd1ba60]]></shop_id>
    <basic_materials>
      <material_bn><![CDATA[testC003]]></material_bn>
      <material_name><![CDATA[testC003]]></material_name>
      <store>10</store>
      <store_freeze>2</store_freeze>
      <weight>0.000</weight>
      <cost>0.000</cost>
      <retail_price>0.000</retail_price>
      <barcode><![CDATA[testC003]]></barcode>
    </basic_materials>
  </lists>
  <count>40</count>
</response>
```

json数据，例：

失败样式：

```
{
  response:{
    "list":[
      {"sales_material_bn":"testC003","sales_material_name":"testC003","unit":"袋","
sales_material_type":"1","retail_price":"12.000","shop_id":"c0b540134794156ebe6ed58cedd1b
a60","basic_materials":[{"material_bn":"testC003","material_name":"testC003","store":"10"
,"store_freeze":"2","weight":"0.000","cost":"0.000","retail_price":"0.000","barcode":"tes
tC003"}]}
    ],
    "count":40
  }
}
```

appropriation.getList 调拨单信息和出入库明细列表

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

根据出入库时间来查询该时间段内的出入库明细

应用级参数

参数名称	是否必要	类型	描述
appropriation_no	否	string	调拨单编号
start_time	是	date	开始时间 2012-12-08 18:50:30
end_time	是	date	结束时间(同上)
page_no	否	number	页码, 默认1 第一页
page_size	否	number	每页数量, 最大100

返回结果

注意: 已经做过特殊字符处理的字段如下: appropriation_no, iostock_bn, branch_bn, branch_name, bn, name, type, memo, original_bn

XML数据, 例:

成功

```
<response>
  <count>1</count>
  <products_count>10</products_count>
  <lists>
    <appropriation_no><![CDATA[S201512181125204]]></appropriation_no>
    <products>
      <bn><![CDATA[fshwxhh]]></bn>
      <name><![CDATA[防水户外鞋]]></name>
      <barcode><![CDATA[fshwxtxm]]></barcode>
      <nums>10</nums>
    </products>
    <iostock>
      <iostock_id>1450410351183999</iostock_id>
      <iostock_bn><![CDATA[R20151218000001]]></iostock_bn>
      <branch_bn><![CDATA[ckbh03]]></branch_bn>
      <branch_name><![CDATA[仓库名称03]]></branch_name>
      <type><![CDATA[调拨出库]]></type>
      <iostock_time><![CDATA[2015-12-18 11:45:51]]></iostock_time>
      <memo>1450409120</memo>
      <original_bn><![CDATA[R20151218000002]]></original_bn>
      <iostock_price>0.000</iostock_price>
      <unit_cost>0.000</unit_cost>
      <iostock_id>1450410549861999</iostock_id>
      <iostock_bn><![CDATA[T20151218000001]]></iostock_bn>
      <branch_bn><![CDATA[ckbh04]]></branch_bn>
      <branch_name><![CDATA[仓库名称04]]></branch_name>
      <type><![CDATA[调拨入库]]></type>
      <iostock_time><![CDATA[2015-12-18 11:49:09]]></iostock_time>
      <memo>1450410351</memo>
      <original_bn><![CDATA[T20151218000001]]></original_bn>
      <iostock_price>0.000</iostock_price>
      <unit_cost>0.000</unit_cost>
    </iostock>
  </lists>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "count": 1,
    "products_count": 10,
    "lists": [
      {
        "appropriation_no": "S201512181125204",
        "products": [
          {
            "bn": "fshwxhh",
            "name": "防水户外鞋",
            "barcode": "fshwxtxm",
            "nums": "10"
          }
        ],
        "iostock": [
          {
            "iostock_id": "1450410351183999",
            "iostock_bn": "R201512180000001",
            "branch_bn": "ckbh03",
            "branch_name": "仓库名称03",
            "type": "调拨出库",
            "iostock_time": "2015-12-18 11:45:51",
            "memo": "1450409120",
            "original_bn": "R201512180000002",
            "iostock_price": "0.000",
            "unit_cost": "0.000"
          },
          {
            "iostock_id": "1450410549861999",
            "iostock_bn": "T201512180000001",
            "branch_bn": "ckbh04",
            "branch_name": "仓库名称04",
            "type": "调拨入库",
            "iostock_time": "2015-12-18 11:49:09",
            "memo": "1450410351",
            "original_bn": "T201512180000001",
            "iostock_price": "0.000",
            "unit_cost": "0.000"
          }
        ]
      }
    ]
  }
}
```


transfer.getList 调拨单信息和出入库明细列表

- [详细描述](#)
- [应用级参数](#)
- [返回结果](#)

详细描述

根据出入库时间来查询该时间段内的出入库明细

应用级参数

参数名称	是否必要	类型	描述
original_bn	否	string	原始单据号
supplier_bn	否	string	供应商编号
branch_bn	否	string	仓库编号
t_type	否	string	出入库类型 E – 直接入库 A – 直接出库 G – 赠品入库 F – 赠品出库 K – 样品入库 J – 样品出库 Y – 分销入库 Z – 分销出库
start_time	是	date	开始时间 2012-12-08 18:50:30
end_time	是	date	结束时间(同上)
page_no	否	number	页码, 默认1 第一页
page_size	否	number	每页数量, 最大100

返回结果

注意: 已经做过特殊字符处理的字段如下: iostock_bn, branch_bn, branch_name, bn, name, type, memo, original_bn

XML数据, 例:

成功

```
<response>
  <count>2</count>
  <iostock_price_num>0.000</iostock_price_num>
  <unit_cost_num>0.000</unit_cost_num>
  <lists>
    <iostock_id>1457495077156233</iostock_id>
    <iostock_bn><![CDATA[Y201603090000001]]></iostock_bn>
    <branch_bn><![CDATA[ckbh04]]></branch_bn>
    <branch_name><![CDATA[仓库名称04]]></branch_name>
    <bn><![CDATA[nk01]]></bn>
    <name><![CDATA[耐克跑鞋]]></name>
    <barcode><![CDATA[nk01]]></barcode>
    <nums>80</nums>
    <type><![CDATA[分销入库]]></type>
    <iostock_time><![CDATA[2016-03-09 11:44:37]]></iostock_time>
    <memo><![CDATA[]]></memo>
    <original_bn><![CDATA[Y201603090000001]]></original_bn>
    <iostock_price>0.000</iostock_price>
    <unit_cost>0.000</unit_cost>
    <iostock_id>145750863084842</iostock_id>
    <iostock_bn><![CDATA[Y201603090000002]]></iostock_bn>
    <branch_bn><![CDATA[ckbh04]]></branch_bn>
    <branch_name><![CDATA[仓库名称04]]></branch_name>
    <bn><![CDATA[nk01]]></bn>
    <name><![CDATA[耐克跑鞋]]></name>
    <barcode><![CDATA[nk01]]></barcode>
    <nums>80</nums>
    <type><![CDATA[分销入库]]></type>
    <iostock_time><![CDATA[2016-03-09 15:30:30]]></iostock_time>
    <memo><![CDATA[]]></memo>
    <original_bn><![CDATA[Y201603090000002]]></original_bn>
    <iostock_price>0.000</iostock_price>
    <unit_cost>0.000</unit_cost>
  </lists>
</response>
```

json数据，例：

成功

```
{
  "response": {
    "count": "2",
    "iostock_price_num": "0.000",
    "unit_cost_num": "0.000",
    "lists": [
      {
        "iostock_id": "1457495077156233",
        "iostock_bn": "Y201603090000001",
        "branch_bn": "ckbh04",
        "branch_name": "仓库名称04",
        "bn": "nk01",
        "name": "耐克跑鞋",
        "barcode": "nk01",
        "nums": "80",
        "type": "分销入库",
        "iostock_time": "2016-03-09 11:44:37",
        "memo": "",
        "original_bn": "Y201603090000001",
        "iostock_price": "0.000",
        "unit_cost": "0.000"
      },
      {
        "iostock_id": "145750863084842",
        "iostock_bn": "Y201603090000002",
        "branch_bn": "ckbh04",
        "branch_name": "仓库名称04",
        "bn": "nk01",
        "name": "耐克跑鞋",
        "barcode": "nk01",
        "nums": "80",
        "type": "分销入库",
        "iostock_time": "2016-03-09 15:30:30",
        "memo": "",
        "original_bn": "Y201603090000002",
        "iostock_price": "0.000",
        "unit_cost": "0.000"
      }
    ]
  }
}
```

仓储接口

为商派ERP仓储接口的调用提供说明.

数据接口版本更新记录

当前文档版本 **Version 1.0**

Date	Version	Description
2015.07.09	1.0	SHOPEX仓储接口文档发布

授权及签名算法

授权

ERP会每个仓储颁发一个wms_node_id, 标识唯一身份

签名算法

- 将所有请求参数（key, value 为一组），对数据结构按照key的升序，重新排序
 - 需要对null值进行过滤
 - 需要将boolean型进行转换为1和0
- 将排序后参数组合为新的字符串（keyvaluekeyvalue 中间无分割符）
- 将生成好的字符串进行MD5加密
- 将新的字符串，统一为大写字符串
- 将生成好的字符串加上“私钥”，形成新的字符串（私钥为ERP系统里面生成）
- 再进行一次MD5加密，得到加密后的字符串
- 再统一为大写字符串，得到最终的签名字符串

php 示例：

```
$private_key = '123456';
static function gen_sign($params){
return strtoupper(md5(strtoupper(md5(self::assemble($params))).$this->private_key));}
static function assemble($params){
    if(!is_array($params)) return null;
    ksort($params, SORT_STRING);
    $sign = '';
    foreach($params AS $key=>$val){
        if(is_null($val)) continue;
        if(is_bool($val)) $val = ($val) ? 1 : 0;
        $sign .= $key . (is_array($val) ? self::assemble($val) : $val);
    }
    return $sign;
}
```

ERP调用WMS接口

调用说明

- 调用方式：POST
- WMS接口地址：<http://@webservice@>
- WMS应用级的参数在绑定的时候需要在ERP里保存，比如：私钥 默认情况下，ERP会给每个WMS配备一个私钥

数据结构

- [WmsOrderItem](#)
- [WmsOrderStatus](#)
- [WmsOrderSourcePlatform](#)
- [WmsInvoiceInfo](#)
- [WmsInvoiceItem](#)
- [PrintRemark](#)
- [WmsItem](#)

WmsOrderItem

名称	类型	示例	描述
item_code	string		商品编码
item_name	string		商品名称
item_quantity	string		商品数量
item_price	string		商品原始价格
item_sale_price	string		商品销售价格
item_id	string		商品Id
item_remark	string		商品备注
item_line_num	string		订单商品列表中商品的行项目编号，即第n行或第n个商品
trade_code	string		可选(若是淘宝交易订单，并且不是赠品，必须要传订单来源编号)
sub_trade_code	string		可选,淘宝子交易号(若是淘宝交易订单,必填)
is_gift	string		判断是否为赠品0:不是1:是
inventory_type	string		库存类型 1可销售库存 101类型用来定义残次品 201冻结类型 库存 301在途库存

WmsOrderStatus

名称	类型	示例	描述
WMS_NEW	String		未开始处理
WMS_PROCESS	String		处理中
WMS_ACCEPT	String		已接单
WMS_PRINT	String		已打印
WMS_PICK	String		已捡货
WMS_CHECK	String		已复核
WMS_PACKAGE	String		已打包
WMS_DELIVERED	String		已发货

WMS_INVOICED	String		已开发票
WMS_RETURNED	String		已退货
WMS_CANCELED	String		取消
WMS_CLOSED	String		关闭
WMS_REJECTED	String		拒收
WMS_FAILURE	String		接单失败
WMS_FINISHED	String		完成（如果确认单据已经完成,推荐使用完成状态）
WMS_PART_IN	String		部分入库
WMS_PART_OUT	String		部分出库
WMS_EXCEPTION	String		异常
WMS_OTHER	String		其他

WmsOrderSourcePlatform

名称	类型	示例	描述
TAOBAO	String		淘宝
PAIPAI	String		拍拍
360BUY	String		京东
SHOPEX_B2B	String		B2B分销
ECOS.B2C	String		Ecstore
SHOPEX_WMALL	String		无线商城
SHOPEX.B2C	String		shopex.b2c
OTHER	String		其他

WmsInvoiceInfo

名称	类型	示例	描述
name	String		名称
phone	String		电话
address	String		地址
taxpayer_id	String		纳税人识别号
bank_name	String		开户银行
bank_account	String		银行账号

WmsInvoiceItem

名称	类型	示例	描述

name	String		物品名称
spec	String		规格型号
unit	String		计量单位
quantity	String		数量
price	String		单价
tax_rate	String		税率
tax_amount	String		税额

PrintRemark

名称	类型	示例	描述
out_date	Date		出库时间 格式:yyyy-MM-dd hh:mm:ss
pay_type	String		支付方式(中文名)
delivery_site	String		配送站点名称
delivery_center	String		配送中心名称
create_date	Date		订购时间 格式:yyyy-MM-dd hh:mm:ss
sortation_code	String		分拣代码

WmsItem

名称	类型	示例	描述
name	String		商品名称
title	String		商品标题
item_code	String		商品编码
remark	String		商品备注
type	String		NORMAL普通商品 COMBINE组合商品 DISTRIBUTION— 分销(默认: NORMAL)
is_sku	String		是否sku (1是 0否) 默认1
gross_weight	String		商品毛重
net_weight	String		商品净重
tare_weight	String		商品皮重
is_friable	String		是否易碎品
weight	String		商品重量
length	String		商品长度
width	String		商品宽度
height	String		商品高度

volume	String	商品体积
property	String	商品属性串 如 颜色:绿色;尺寸:23;
pricing_cat	String	计价货类
package_material	String	商品包装材料类型
price	String	商品价格
barcode	String	商品条形码
unit	String	商品单位
package_spec	String	商品包装规格
ename	String	商品英文名称
brand	String	商品品牌
batch_no	String	批次号
support_expire_date	String	是否进行保质期管理（1为进行保质期管理，0为不进行保质期管理）默认0
expire_date	String	保质期到期日（默认当前日期）
support_barcode	String	是否进行条形码管理（00为不扫，01为出库扫，10为入库扫，11为出入都扫）默认00
support_antifake	String	是否进行防伪码管理（1是 0否）默认0
antifake_code	String	商品防伪码
slave_product_code	String	客户商品编号（与一般商品相同，此处记录从属关系），格式为多个子商品用英文逗号隔开
goods_cat	String	商品分类
color	String	商品颜色

接口列表

入库单创建

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.inorder.create	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
uniqid	string		请求标识(在ISV即请求方系统中全局唯一),用于控制并发及去重
out_order_code	string		客户入库单号
order_type	string		订单类型：采购入库: IN_PURCHASE, 调拨入库: IN_ALLCOATE, 调帐入库: IN_ADJUSTMENT, 转储入库: IN_DUMP, 换货入库: IN_EXCHANGE, 残损入库: IN_DEFECTIVE, 其他入库: IN_OTHER
order_source	WmsOrderSourcePlatform		订单来源平台
logistics_code	string		运输公司编号
logistics_no	string		运输公司运单号
remark	string		备注
created	Date		创建时间
wms_order_code	string		wms订单编号
is_finished	string		该订单是否已完成，如果完成则设置为true，如果为false，则需要等待继续创建订单商品信息。

current_page	String		页码
total_page	String		总页码
expect_start_time	Date		期望开始送达时间
expect_end_time	Date		期望结束送达时间
shipper_name	string		发货人姓名
shipper_zip	string		发货人邮编
shipper_state	string		省
shipper_city	string		市
shipper_district	string		区
shipper_address	string		地址
shipper_phone	string		电话
shipper_mobile	string		手机
shipper_email	string		邮箱
total_goods_fee	string		总额
line_total_count	string		行项目总数
item_total_num	string		订单商品明细总数量
storage_code	string		库内存放点编号
items	WmsOrderItem		商品信息明细列表（json字符串）

请求参数示例

```
[{
  "uniqid": "69bf6b7bc37cd72d2e40e22e1ed395a5",
  "out_order_code": "I201507101517008642",
  "order_type": "IN_PURCHASE",
  "created": "2015-07-10 15:11:30",
  "wms_order_code": "I201507101517008642",
  "logistics_code": "",
  "logistics_no": "",
  "remark": null,
  "shipper_name": "我是供应商",
  "shipper_zip": "未知",
  "shipper_state": "未知",
  "shipper_city": "未知",
  "shipper_district": "未知",
  "shipper_address": "未知",
  "shipper_phone": "未知",
  "shipper_mobile": "未知",
  "shipper_email": "未知",
  "total_goods_fee": "9900.000",
  "storage_code": "D1",
  "items": "{\"item\": [{\"item_code\": \"test1\", \"item_name\": \"\\u6d4b\\u8bd5\\u5546\\u54c1\\u4e00\", \"item_quantity\": \"100\", \"item_price\": \"99.000\", \"item_line_num\": 1, \"trade_code\": \"\", \"item_id\": \"test1\", \"is_gift\": \"0\", \"item_remark\": \"\", \"inventory_type\": \"1\"}]}",
  "is_finished": "true",
  "current_page": 1,
  "total_page": 1,
  "item_total_num": 1,
  "line_total_count": 1,
  "callback_url": "http://erp.trunk.localhost/index.php/openapi/asynccallback/async_result_handler/id/14365201534491330671339-1436520153",
  "task": "14365201534491330671339",
  "v": "1.0",
  "method": "store.wms.inorder.create",
  "format": "json",
  "timestamp": 1436520153,
  "wms_node_id": "o1111725941",
  "sign": "0a32bb8fc350ef7acc7e88743f672fff"
}]
```

响应结果

json格式:

```
{
  "rsp": "succ",
  "data": {
    "wms_order_code": "LP31209123123",
    "created": "2011-08-16 19:27:29"
  }
}
```


查询入库单状态(同步)

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.inorder.get	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
out_order_code	string		客户订单编号，即相对于WMS系统的外部订单号
order_type	string		订单类型：采购入库: IN_PURCHASE, 调拨入库: IN_ALLCOATE, 调帐入库: IN_ADJUSTMENT, 转储入库: IN_DUMP, 换货入库: IN_EXCHANGE, 残损入库: IN_DEFECTIVE, 其他入库: IN_OTHER

请求参数示例

```
[{
  "out_order_code": "wms123456789",
  "v": "1.0",
  "method": "store.wms.inorder.get",
  "format": "json",
  "timestamp": 1436520530,
  "wms_node_id": "o1111725941",
  "sign": "a743fa1991bb0bdbfa203716fd01101c"
}]
```

响应结果

名称	类型	示例	描述
wms_order_code	string		第三方仓储订单编号
out_order_code	string		商家订单编号
status_code	WmsOrderStatus		

json格式:

```
{
  "rsp": "succ",
  "data": {
    "status_code": 'FINISHED'
  }
}
```

取消入库单(同步)

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.inorder.cancel	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
out_order_code	string		客户订单编号, 即相对于WMS系统的外部订单号
order_type	string		订单类型: 采购入库: IN_PURCHASE, 调拨入库: IN_ALLCOATE, 调帐入库: IN_ADJUSTMENT, 转储入库: IN_DUMP, 换货入库: IN_EXCHANGE, 残损入库: IN_DEFECTIVE, 其他入库: IN_OTHER

请求参数示例

```
[{
  "out_order_code": "wms123456789",
  "v": "1.0",
  "method": "store.wms.inorder.get",
  "format": "json",
  "timestamp": 1436520658,
  "wms_node_id": "o1111725941",
  "sign": "7b3fc08b19a407f66f16ae55d7cb36e0"
}]
```

响应结果

名称	类型	示例	描述
modified	string		修改时间

json格式:

```
{
  "rsp": "succ",
  "msg_id": "3463421b2581604c74dcb9b74ffab2e4"
  "data": {
    "is_success": true,
    "modified": "2011-07-21 20:35:15",
  }
}
```


创建出库单

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.outorder.create	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
uniqid	string		请求标识(在ISV即请求方系统中全局唯一),用于控制并发及去重
out_order_code	string		出库单号
order_type	string		订单类型: 调拨出库: OUT_ALLCOATE, 调帐出库: OUT_ADJUSTMENT, 残损出库: OUT_DEFECTIVE, 采购退货出库: OUT_PURCHASE_RETURN, 其他出库: OUT_OTHER
order_source	WmsOrderSourcePlatform		订单来源平台
shipping_type	String		运输方式: EXPRESS-快递 EMS-邮政速递
logistics_code	String		快递公司编号
logistics_no	String		快递单号
remark	String		备注
created	String		单据创建时间
wms_order_code	String		wms 订单编号

is_finished	String	该订单是否已完成，如果完成则设置为true，如果为false，则需要等待继续创建订单商品信息
current_page	String	当前分页数
total_page	String	总分页数
has_invoice	String	是否开发票。可选值：true(开),false(不开)。默认不开
invoice_type	String	发票类型。如果要invoice为true,则此参数为必须
invoice_title	String	发票title。如果要invoice为true,则此参数为必须
invoice_fee	String	发票税金。如果要invoice为true,则此参数为必须
invoice_desc	String	发票内容。如果要invoice为true,则此参数为必须
discount_fee	String	折扣优惠金额
is_protect	String	是否保价（针对物流配送）
protect_fee	String	保价费用（针对物流配送）
is_cod	String	是否货到付款。可选值:true(是),false(否)
cod_fee	String	应收货款（用于货到付款）
cod_service_fee	String	cod服务费（货到付款 必填）
total_goods_fee	String	订单商品总价（精确到小数点后2位）
receiver_name	String	收货人姓名
receiver_zip	String	收货人邮政编码
receiver_phone	String	收货人电话号码（如有分机号用“-”分隔）（电话和手机必选一项）
receiver_mobile	String	收货人手机号码(电话和手机必选一项)
receiver_state	String	收货人所在省，如浙江省、北京等
receiver_city	String	收货人所在市，如杭州市、上海市等
receiver_district	String	收货人所在县（区），注意有些市下面没有区的，如：义乌市
receiver_address	String	收货人的详细地址
receiver_email	String	收货人email地址
receiver_time	String	收货人要求到货时间
sign_memo	String	签收备注(如：本人签收)
storage_code	String	库内存放点编号
line_total_count	String	订单商品明细总行数(即整个单据商品明细行数,而不是当前分页的商品明细行数)
item_total_num	String	订单商品明细总数量

items	WmsOrderItem	商品信息明细列表 (json字符串)
-------	--------------	--------------------

请求参数示例

```
[{
  "uniqid": "32c7694c0c06b6b59c1b1bab0378638f",
  "customer_id": "",
  "out_order_code": "H201507101717004061",
  "order_type": "OUT_PURCHASE_RETURN",
  "created": "2015-07-10 17:37:33",
  "wms_order_code": "H201507101717004061",
  "shipping_type": "EXPRESS",
  "logistics_code": "",
  "remark": null,
  "total_amount": "99.000",
  "receiver_name": "未知",
  "receiver_zip": "123211",
  "receiver_state": "未知",
  "receiver_city": "未知",
  "receiver_district": "未知",
  "receiver_address": "未知",
  "receiver_phone": "未知",
  "receiver_mobile": "未知",
  "receiver_email": "未知",
  "receiver_time": "",
  "sign_standard": "",
  "source_plan": "",
  "storage_code": "D1",
  "items": "{\"item\": [{\"item_code\": \"test1\", \"item_name\": \"\\u6d4b\\u8bd5\\u5546\\u54c1\\u4e00\", \"item_quantity\": \"1\", \"item_price\": \"99.000\", \"item_line_num\": 1, \"trade_code\": \"\", \"item_id\": \"test1\", \"is_gift\": \"0\", \"item_remark\": \"\", \"inventory_type\": \"1\"}]}",
  "is_finished": "true",
  "current_page": 1,
  "total_page": 1,
  "item_total_num": 1,
  "line_total_count": 1,
  "callback_url": "http://erp.trunk.localhost/index.php/openapi/asynccallback/async_result_handler/id/143652106971351773013270-1436521069",
  "task": "143652106971351773013270",
  "v": "1.0",
  "method": "store.wms.outorder.create",
  "format": "json",
  "timestamp": 1436521069,
  "wms_node_id": "o1111725941",
  "sign": "472071a2432e783689484166657672e1"
}]
```

响应结果

名称	类型	示例	描述
----	----	----	----

wms_order_code	string		第三方仓储的订单编号
created	string		订单创建时间

json格式:

```
{
  "rsp": "succ",
  "data": {
    "wms_order_code": "LP31209123123",
    "created": "2011-08-16 19:27:29"
  }
}
```

取消出库单(同步)

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.outorder.cancel	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
out_order_code	string		客户订单编号, 即相对于WMS系统的外部订单号
order_type	string		订单类型: 调拨出库: OUT_ALLCOATE, 调帐出库: OUT_ADJUSTMENT, 残损出库: OUT_DEFECTIVE, 采购退货出库: OUT_PURCHASE_RETURN, 其他出库: OUT_OTHER

请求参数示例

```
[{
  "out_order_code": "H201507101717004061",
  "order_type": "OUT_PURCHASE_RETURN",
  "v": "1.0",
  "method": "store.wms.outorder.cancel",
  "format": "json",
  "timestamp": 1436521313,
  "wms_node_id": "o1111725941",
  "sign": "850ed6877080ddc0b92d4958ef62b885"
}]
```

响应结果

名称	类型	示例	描述
modified	Date		修改时间

json格式:

```
{
  "rsp": "succ",
  "data": {
    "is_success": true,
    "modified": "2011-07-21 20:35:15",
  }
}
```

出库单状态查询(同步)

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.outorder.get	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
out_order_code	String		客户订单编号, 即相对于WMS系统的外部订单号
order_type	String		订单类型: 调拨出库: OUT_ALLCOATE, 调帐出库: OUT_ADJUSTMENT, 残损出库: OUT_DEFECTIVE, 采购退货出库: OUT_PURCHASE_RETURN, 其他出库: OUT_OTHER

请求参数示例

```
[{
  "out_order_code": "H201507101717001814",
  "order_type": "OUT_PURCHASE_RETURN",
  "v": "1.0",
  "method": "store.wms.outorder.cancel",
  "format": "json",
  "timestamp": 1436522252,
  "wms_node_id": "o1111725941",
  "sign": "c98721c8a0c05e9046ab7c485f2cff0f"
}]
```

响应结果

名称	类型	示例	描述
wms_order_code	String		第三方仓储订单编号
out_order_code	String		商家订单编号
status_code	WmsOrderStatus		订单状态

json格式:

```
{
  "rsp": "succ",
  "data": {
    "status_code": 'WMS_FINISHED'
  }
}
```


创建转储单

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.transferorder.create	方法
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	示例	描述
uniqid	String		客户订单编号, 即相对于WMS系统的外部订单号
out_order_code	String		转储单号
remark	String		备注
created	String		创建时间
is_finished	String		该订单是否已完成, 如果完成则设置为true, 如果为false, 则需要等待继续创建订单商品信息。
current_page	String		当前分页数
total_page	String		总分页数
src_storage	String		来源存放点编号
dest_storage	String		目的存放点编号
line_total_count	String		订单商品明细总行数(即整个单据商品明细行数,而不是当前分页的商品明细行数)
items	WmsOrderItem		商品信息明细列表(json字符串)

请求参数示例

```
[{
  "uniqid": "bd1db064293bd283c42f8395c392e933",
  "original_id": "",
  "out_order_code": "20150710000001",
  "created": "2015-07-10 18:00:13",
  "remark": "",
  "src_storage": "D1",
  "dest_storage": "C1",
  "line_total_count": 1,
  "items": "{\"item\": [{\"item_code\": \"test1\", \"item_name\": \"\\u6d4b\\u8bd5\\u5546\\u54c1\\u4e00\", \"item_quantity\": \"1\", \"item_price\": \"0.000\", \"item_line_num\": 1, \"item_remark\": \"\"}]}",
  "is_finished": "true",
  "current_page": 1,
  "total_page": 1,
  "callback_url": "http://erp.trunk.localhost/index.php/openapi/asynccallback/async_result_handler/id/14365224138163420138188-1436522413",
  "task": "14365224138163420138188",
  "v": "1.0",
  "method": "store.wms.transferorder.create",
  "format": "json",
  "timestamp": 1436522413,
  "wms_node_id": "o1111725941",
  "sign": "a81c1013fc284080c0ebf7ef2a1fe314"
}]
```

响应结果

名称	类型	是否必须	示例	描述
wms_order_code	String	否		第三方仓储订单编号
created	Date	否		修改时间

json格式:

```
{
  "rsp": "succ",
  "data": {
    "wms_order_code": "LP31209123123",
    "created": "2011-08-16 19:27:29"
  }
}
```

取消转储单(同步)

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.transferorder.cancel	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
out_order_code	String		客户订单编号，即相对于WMS系统的外部订单号

请求参数示例

```
[{
  "out_order_code": "20150710000001",
  "callback_url": "",
  "task": "14365224138163420138188",
  "v": "1.0",
  "method": "store.wms.transferorder.create",
  "format": "json",
  "timestamp": 1436522413,
  "wms_node_id": "o1111725941",
  "sign": "a81c1013fc284080c0ebf7ef2a1fe314"
}]
```

响应结果

名称	类型	是否必须	示例	描述
----	----	------	----	----

modified	Date	否		修改时间
----------	------	---	--	------

json格式:

```
{
  "rsp": "succ",
  "msg_id": "3463421b2581604c74dcb9b74ffab2e4"
  "data": {
    "is_success": true,
    "modified": "2011-07-21 20:35:15",
  }
}
```

同步供应商

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.vendors.get	方法
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	示例	描述
CustomerID	String		供应商编号
vendor_name	String		供应商简称
address	String		地址
state	String		省
city	String		市
country	String		国家

请求参数示例

```
[{
  "CustomerID": "STO",
  "vendor_ename": "我是供应商",
  "vendor_name": "我是供应商",
  "address": "",
  "state": null,
  "city": null,
  "country": "中国",
  "v": "1.0",
  "method": "store.vendors.get",
  "format": "json",
  "timestamp": 1436522637,
  "wms_node_id": "o1111725941",
  "sign": "c7765899fd66f2dc26341c3adf355e62"
}]
```

响应结果

json格式:

```
{
  "rsp": "succ",
  "data": {}
}
```

创建发货单

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.saleorder.create	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
uniqid	String		请求标识(在ISV即请求方系统中全局唯一),用于控制并发及去重
out_order_code	WmsOrderSourcePlatform		订单来源平台
platform_order_code	String		来源平台单号
shipping_type	String		运输方式: EXPRESS-快递 EMS-邮政 速递
shipping_fee	Price		配送费用
logistics_code	String		快递公司编号
logistics_no	String		快递单号
shop_code	String		店铺编号/售达方编号
remark	String		备注
created	String		单据创建时间
wms_order_code	String		wms订单编号
is_finished	Boolean		该订单是否已完成, 如果完成则设置为true, 如果为false, 则需要等待继

			续创建订单商品信息。
current_page	Number		当前分页数
total_page	Number		total_page
has_invoice	String		是否开发票。可选值： true(开),false(不开)。默认不开
invoice_type	String		发票类型。如果要invoice为true,则此参数为必须。可选值：general（普通发票）,vat（增值税发票）,special（专用发票）
invoice_title	String		发票title。如果要invoice为true,则此参数为必须
invoice_issued_id	String		开票方编号
invoice_fee	String		发票税金。 如果要invoice为true,则此参数为必须
invoice_info	WmsInvoiceInfo		发票信息(json串)。增值税发票类型必填
invoice_item	WmsInvoiceItem		发票明细（json串列表）。 invoice_item和invoice_desc二者必选一项（优先取invoice_item）
invoice_desc	String		发票内容
discount_fee	Price		订单折扣优惠金额
is_protect	Boolean		是否保价（针对物流配送）
protect_fee	Price		保价费用（针对物流配送）
is_cod	Boolean		是否货到付款。可选值:true(是),false(否)
cod_fee	Price		应收货款（用于货到付款）
cod_service_fee	Price		cod服务费（货到付款 必填）
total_goods_fee	Price		订单商品总价
total_trade_fee	Price		订单交易金额（包括商品价格 + 配送费用 精确到小数点后2位）
receiver_name	String		收货人姓名
receiver_zip	String		收货人邮政编码
receiver_phone	String		收货人电话号码（如有分机号用“-”分隔）(电话和手机必选一项)
receiver_mobile	String		收货人手机号码(电话和手机必选一项)
receiver_state	String		收货人所在省，如浙江省、北京等
receiver_city	String		收货人所在市，如杭州市、上海市等
receiver_district	String		收货人所在县（区），注意有些市下面是没有区的，如：义乌市
receiver_address	String		收货人的详细地址
receiver_email	String		收货人email地址

receiver_time	Date	收货人要求到货时间
sign_memo	String	
storage_code	String	库内存放点编号
line_total_count	Number	订单商品明细总行数(即整个单据商品明细行数,而不是当前分页的商品明细行数)
item_total_num	Number	订单商品明细总数量
items	WmsOrderItem	商品信息明细列表(json字符串)
print_remark	PrintRemark	面单打印内容字段。(目前只支持京东业务)
dispatch_notify	String	配送前是否通知。true 为通知; false 为不通知
dispatch_time	Date	配送时间

请求参数示例

```
[{
  "uniqid": "21d3b473af2dc74b3c26a119ef9e86b9",
  "out_order_code": "1507101100002",
  "order_source": "OTHER",
  "shipping_type": "EXPRESS",
  "shipping_fee": 0,
  "platform_order_code": "201507101810000486",
  "logistics_code": "EMS",
  "shop_code": "taobao",
  "remark": "",
  "created": "2015-07-10 18:08:14",
  "wms_order_code": "1507101100002",
  "is_finished": "true",
  "current_page": 1,
  "total_page": 1,
  "has_invoice": "false",
  "invoice_type": null,
  "invoice_title": null,
  "invoice_fee": null,
  "invoice_info": null,
  "invoice_desc": null,
  "invoice_item": null,
  "discount_fee": 0,
  "is_protect": "false",
  "protect_fee": "0.000",
  "is_cod": "true",
  "cod_fee": 199,
  "cod_service_fee": "0",
  "total_goods_fee": 199,
  "total_trade_fee": 199,
  "receiver_name": "陈平",
  "receiver_zip": "200003",
  "receiver_phone": "201 51086858"
```

```

    receiver_phone : 201-51088858 ,
    "receiver_mobile": "",
    "receiver_state": "上海",
    "receiver_city": "上海市",
    "receiver_district": "徐汇区",
    "receiver_address": "桂林路396号",
    "receiver_email": "",
    "receiver_time": "任意日期 任意时间段",
    "line_total_count": 1,
    "item_total_num": 1,
    "storage_code": "D1",
    "items": "{\n"item": [{\n"item_code":\n"test1",\n"item_name":\n"\u6d4b\u8bd5\u5546\u54c1\u4e00",\n"item_quantity":\n"1",\n"item_price":199,\n"item_line_num":1,\n"trade_code":\n"201507101810000486",\n"item_id":\n"test1",\n"is_gift":\n"0",\n"item_remark":null,\n"inventory_type":\n"1",\n"item_sale_price":199}}],
    "print_remark": "",
    "dispatch_time": null,
    "callback_url": "http://erp.trunk.localhost/index.php/openapi/asynccallback/async_result_handler/id/143652289523191822590169-1436522895",
    "task": "143652289523191822590169",
    "v": "1.0",
    "method": "store.wms.saleorder.create",
    "format": "json",
    "timestamp": 1436522895,
    "wms_node_id": "o1111725941",
    "sign": "742d77e6a9ed1db29a976e1ed872565f"
  }
}

```

响应结果

名称	类型	是否必须	示例	描述
wms_order_code	String	否		第三方仓储的订单编号
created	Date	否		订单创建时间

json格式:

```

{
  "rsp": "succ",
  "data": {
    "wms_order_code": "LP31209123123",
    "created": "2011-08-16 19:27:29"
  }
}

```

取消发货单(同步)

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.saleorder.cancel	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
out_order_code	String		客户订单编号，即相对于WMS系统的外部订单号

请求参数示例

```
[{
  "out_order_code": "d123456789",
  "v": "1.0",
  "method": "store.vendors.get",
  "format": "json",
  "timestamp": 1436522637,
  "wms_node_id": "o1111725941",
  "sign": "c7765899fd66f2dc26341c3adf355e62"
}]
```

响应结果

名称	类型	是否必须	示例	描述
modified	Date	否		修改时间

json格式:

```
{
  "rsp": "succ",
  "data": {
    "is_success": true,
    "modified": "2011-07-21 20:35:15",
  }
}
```

添加物料

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.item.add	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
item_lists	WmsItem		商品列表(json字符串)

请求参数示例

```
[{
  "item_lists": "{\\"item\\":[{\\"name\\":null,\\"title\\":null,\\"item_code\\":\\"test1\\",\\"remark\\":\\"\\",\\"type\\":\\"NORMAL\\",\\"is_sku\\":\\"1\\",\\"gross_weight\\":\\"\\",\\"net_weight\\":\\"\\",\\"tare_weight\\":\\"\\",\\"is_friable\\":\\"\\",\\"is_dangerous\\":\\"\\",\\"pricing_cat\\":\\"\\",\\"package_material\\":\\"\\",\\"price\\":\\"\\",\\"support_batch\\":\\"\\u5426\\",\\"support_expire_date\\":\\"\\u5426\\",\\"expire_date\\":\\"2015-07-10\\",\\"support_barcode\\":\\"0\\",\\"barcode\\":\\"\\",\\"support_antifake\\":\\"\\u5426\\",\\"unit\\":\\"\\",\\"package_spec\\":\\"\\",\\"ename\\":\\"\\",\\"brand\\":\\"\\",\\"batch_no\\":\\"\\",\\"goods_cat\\":\\"\\",\\"color\\":\\"\\",\\"property\\":\\"\\"}]}",
  "uniqid": "6dfdb40cbf0fa8efc04a6abf5c2dbf2c",
  "callback_url": "http://:/index.php/openapi/asynccallback/async_result_handler/id/143652315856891484156271-1436523158",
  "task": "143652315856891484156271",
  "v": "1.0",
  "method": "store.wms.item.add",
  "format": "json",
  "timestamp": 1436523158,
  "wms_node_id": "o1111725941",
  "sign": "bdea44560603851491d194fbac65ade8"
}]
```

响应结果

名称	类型	是否必须	示例	描述
item_code	String	是		商家编码
wms_item_code	String	是		第三方仓储商品编码

json格式:

```
{
  "rsp":"fail",
  "data":{
    'succ': [
      {
        'item_code': '123456',
        'wms_item_code': 'w1b987654321'
      }
    ],
    'error': [
      {
        'item_code': '654321',
        'wms_item_code': '',
        "error_code": 'xxxx',
        "error_description": '编号重复'
      }
    ]
  }
}
```

更新物料

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.item.update	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
item_lists	WmsItem		商品列表(json字符串)

请求参数示例

```
[{
  "item_lists": "{\\"item\\":[{\\"name\\":null,\\"title\\":null,\\"item_code\\":\\"test1\\",\\"remark\\":\\"\\",\\"type\\":\\"NORMAL\\",\\"is_sku\\":\\"1\\",\\"gross_weight\\":\\"\\",\\"net_weight\\":\\"\\",\\"tare_weight\\":\\"\\",\\"is_friable\\":\\"\\",\\"is_dangerous\\":\\"\\",\\"pricing_cat\\":\\"\\",\\"package_material\\":\\"\\",\\"price\\":\\"\\",\\"support_batch\\":\\"\\u5426\\",\\"support_expire_date\\":\\"\\u5426\\",\\"expire_date\\":\\"2015-07-10\\",\\"support_barcode\\":\\"0\\",\\"barcode\\":\\"\\",\\"support_antifake\\":\\"\\u5426\\",\\"unit\\":\\"\\",\\"package_spec\\":\\"\\",\\"ename\\":\\"\\",\\"brand\\":\\"\\",\\"batch_no\\":\\"\\",\\"goods_cat\\":\\"\\",\\"color\\":\\"\\",\\"property\\":\\"\\"}]}",
  "uniqid": "6dfdb40cbf0fa8efc04a6abf5c2dbf2c",
  "callback_url": "http://:/index.php/openapi/asynccallback/async_result_handler/id/143652315856891484156271-1436523158",
  "task": "143652315856891484156271",
  "v": "1.0",
  "method": "store.wms.item.add",
  "format": "json",
  "timestamp": 1436523158,
  "wms_node_id": "o1111725941",
  "sign": "bdea44560603851491d194fbac65ade8"
}]
```

响应结果

名称	类型	是否必须	示例	描述
rsp	String	是		显示调用结果
res	String	是		显示调用失败信息
sign	String	是		签名
data	String	是		应用级返回参数

json格式:

```
{
  "rsp": "succ",
  "msg_id": "3463421b2581604c74dcb9b74ffab2e4"
  "data": {
    'succ': [
      {
        'item_code': '123456',
        'wms_item_code': 'w1b987654321'
      }
    ],
    'error': []
  }
}
```


退货单创建

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.returnorder.create	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
uniqid	String		请求标识(在ISV即请求方系统中全局唯一),用于控制并发及去重
out_order_code	String		退货单号
order_source	WmsOrderSourcePlatform		订单来源平台
platform_order_code	String		来源平台单号
orig_order_code	String		原始发货单号
logistics_no	String		快递单号
logistics_code	String		快递公司编号
return_amount	String		退货金额
remark	String		备注
created	Date		单据创建时间
wms_order_code	String		wms订单编号
is_finished	String		该订单是否已完成，如果完成则设置为true，如果为false，则需要等待继续创建订单商品信息。

current_page	String		当前分页数
total_page	String		总分页数
receiver_name	String		退货人姓名
receiver_zip	String		退货人邮政编码
receiver_state	String		退货人所在省，如浙江省、北京等
receiver_city	String		退货人所在市，如杭州市、上海市等
receiver_district	String		退货人所在县（区），注意有些市下面是没有区的，如：义乌市
receiver_address	String		退货人的详细地址
receiver_phone	String		退货人电话号码（如有分机号用“-”分隔）（电话和手机必选一项）
receiver_mobile	String		退货人手机号码（电话和手机必选一项）
receiver_email	String		退货人email地址
storage_code	String		库内存放点编号
line_total_count	String		订单商品明细总行数（即整个单据商品明细行数，而不是当前分页的商品明细行数）
item_total_num	String		订单商品明细总数量
items	WmsOrderItem		商品信息明细列表(json字符串)

请求参数示例

```
[{
  "uniqid": "d361e3f100cb80a63e7e116856d0e81b",
  "wms_supplier": "",
  "out_order_code": "201507101813004458",
  "warehouse_code": "branch2",
  "orig_order_code": "1507081100001",
  "created": "2015-07-10 18:15:00",
  "logistics_no": "",
  "logistics_code": "",
  "remark": "",
  "platform_order_code": "201507080910009448",
  "wms_order_code": "201507101813004458",
  "is_finished": "true",
  "current_page": "1",
  "total_page": "1",
  "receiver_name": "陈平",
  "receiver_zip": "200003",
  "receiver_state": null,
  "receiver_city": "上海市",
  "receiver_district": "徐汇区",
  "receiver_address": null,
  "receiver_phone": null,
  "receiver_mobile": "",
  "receiver_email": "",
  "sign_code": "",
  "dest_plan": "",
  "line_total_count": 2,
  "storage_code": "A2",
  "items": "{ \"item\": [{ \"item_code\": \"test1\", \"item_name\": \"\\u6d4b\\u8bd5\\u5546\\u54c1\\u4e00\", \"item_quantity\": \"1\", \"item_price\": \"199.000\", \"item_line_num\": 1, \"trade_code\": \"\", \"item_id\": \"\", \"is_gift\": \"0\", \"item_remark\": \"\", \"inventory_type\": \"1\" }, { \"item_code\": \"test3\", \"item_name\": \"\\u6d4b\\u8bd5\\u5546\\u54c1\\u4e09\", \"item_quantity\": \"1\", \"item_price\": \"199.000\", \"item_line_num\": 2, \"trade_code\": \"\", \"item_id\": \"\", \"is_gift\": \"0\", \"item_remark\": \"\", \"inventory_type\": \"1\" } ] }",
  "callback_url": "http://index.php/openapi/asynccallback/async_result_handler/id/143652335969041474122842-1436523359",
  "task": "143652335969041474122842",
  "v": "1.0",
  "method": "store.wms.returnorder.create",
  "format": "json",
  "timestamp": 1436523359,
  "wms_node_id": "o1111725941",
  "sign": "586abecc8b669511ba0a0868f18ca934"
}]
```

响应结果

名称	类型	是否必须	示例	描述
wms_order_code	String	否		第三方仓储的订单编号
created	Date	否		订单创建时间

json格式:

```
{
  "rsp": "succ",
  "data": {
    "wms_order_code": "LP31209123123",
    "created": "2011-08-16 19:27:29"
  }
}
```

取消退货单(同步)

- [系统级参数](#)
- [业务级参数](#)
- [请求参数示例](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式，支持： json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	store.wms.returnorder.cancel	方法
timestamp	int	是	1435912385	时间戳
task	String	是	14364344528513665503176	任务号
wms_node_id	String	是		Wms在ERP唯一
callback_url	String	否		回调URL

业务级参数

名称	类型	示例	描述
out_order_code	String		客户订单编号，即相对于WMS系统的外部订单号

请求参数示例

```
[{
  "out_order_code": "201507101813004458",
  "warehouse_code": "branch2",
  "v": "1.0",
  "method": "store.wms.returnorder.cancel",
  "format": "json",
  "timestamp": 1436523426,
  "wms_node_id": "o1111725941",
  "sign": "ae02dedc2b4f871811bcf3b6fbf6834d"
}]
```

响应结果

名称	类型	是否必须	示例	描述

modified	Date	否		修改时间
----------	------	---	--	------

json格式:

```
{
  "rsp": "succ",
  "data": {
    "is_success": true,
    "modified": "2011-07-21 20:35:15",
  }
}
```

WMS调用ERP接口

接口地址

<http://@domain@/openapi/process/handle>

接口列表

wms.stockin.status_update 入库单状态更新

- 系统级参数
- 业务级参数
- items
- 响应结果

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	wms.stockin.status_update	方法
node_id	String	是	3995507852	绑定状态
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	是否必须	示例	描述
stockin_bn	String	是	I201507021313004060	入库单号,首字母必须以I、T、D、O打头。I代表采购入库, T代表采购入库, D代表残损入库, O代表其它入库
warehouse	String	否	wsm_a1	创建编号
io_status	String	是	CANCEL	状态,可选值: PARTIN代表部分入库、FINISH代表全部入库、CANCEL代表取消
remark	String	否	这是备注	备注
operate_time	String	否	2015-07-03 10:10:10	操作时间
item	ITEM	是		明细 需json_encode

items

名称	类型	是否必须	示例	描述
product_bn	String	是	sp123	货号
normal_num	int	是	1	良品
defective_num	int	是	1	不良品

响应结果

json格式:

```
{
  "rsp": "succ",
  "data": "",
  "res": ""
}
```

wms.stockout.status_update 出库单状态更新

- 系统级参数
- 业务级参数
- items
- 响应结果

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	wms.stockout.status_update	方法
node_id	String	是	3995507852	绑定状态
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	是否必须	示例	描述
stockout_bn	String	是	H201507021313004060	出库单号,首字母必须以H、R、B、U打头。H代表采购退货, R代表调拨出库, B代表残损出库, U代表其它出库
warehouse	String	否	wsm_a1	创建编号
io_status	String	是	CANCEL	状态,可选值: PARTIN代表部分出库、FINISH代表全部出库、CANCEL代表取消
remark	String	否	这是备注	备注
operate_time	String	否	2015-07-03 10:10:10	操作时间
item	ITEM	是		明细 需json_encode

items

名称	类型	是否必须	示例	描述
product_bn	String	是	sp123	货号
num	int	是	1	出库数量

响应结果

json格式:

```
{
  "rsp": "succ",
  "data": "",
  "res": ""
}
```

wms.stockdump.status_update 转储单状态更新

- 系统级参数
- 业务级参数
- items
- 响应结果

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	wms.stockdump.status_update	方法
node_id	String	是	3995507852	绑定状态
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	是否必须	示例	描述
stockdump_bn	String	是	201507021313004060	转储单号
warehouse	String	否	wsm_a1	创建编号
io_status	String	是	CANCEL	状态, 可选值: PARTIN代表部分入库、FINISH代表全部入库、CANCEL代表取消
remark	String	否	这是备注	备注
operate_time	String	否	2015-07-03 10:10:10	操作时间
item	ITEM	是		明细 需json_encode

items

名称	类型	是否必须	示例	描述
product_bn	String	是	sp123	货号
num	int	是	1	数量

响应结果

json格式:

```
{  
  "rsp": "succ",  
  "data": "",  
  "res": ""  
}
```

wms.delivery.status_update 发货单状态更新

- 系统级参数
- 业务级参数
- items
- 响应结果

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	wms.delivery.status_update	方法
node_id	String	是	3995507852	绑定状态
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	是否必须	示例	描述
delivery_bn	String	是	201507021313004060	发货单号
warehouse	String	否	wsm_a1	仓库编号
logi_no	String	否	sto123456	运单号
logistics	String	否	EMS	物流公司
status	String	是	CANCEL	状态, 可选值: CLOSE代表取消、ACCEPT代表已接收、PRINT代表已打印、PICK代表拣货、CHECK代表校验、PACKAGE代表打包、DELIVERY代表发货、UPDATE代表更新、
weight	float	否	0	重量
volume	float	否	0	商品体积
remark	String	否	这是备注	备注
operate_time	String	否	2015-07-03 10:10:10	操作时间
out_delivery_bn	String	否	T201507021313004060	外部单号
item	ITEM	是		明细 需json_encode

items

名称	类型	是否必须	示例	描述
product_bn	String	是	sp123	货号
num	int	是	1	数量

响应结果

json格式:

```
{
  "rsp": "succ",
  "data": "",
  "res": ""
}
```

wms.reship.status_update 退货单状态更新

- 系统级参数
- 业务级参数
- items
- 响应结果

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	wms.reship.status_update	方法
node_id	String	是	3995507852	绑定状态
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	是否必须	示例	描述
reship_bn	String	是	201507021313004060	退货单号
logi_no	String	否	sto123456	运单号
logistics	String	否	EMS	物流公司
warehouse	String	否	wsm_a1	仓库编号
status	String	是	CANCEL	状态, 可选值: CLOSE代表取消、DENY代表拒绝、FINISH代表全部退货、PARTIN代表部分退货
remark	String	否	这是备注	备注
operate_time	String	否	2015-07-03 10:10:10	操作时间
item	ITEM	是		明细 需json_encode

items

名称	类型	是否必须	示例	描述
product_bn	String	是	sp123	货号
normal_num	int	是	1	良品
defective_num	int	是	1	不良品

响应结果

json格式:

```
{
  "rsp": "succ",
  "data": "",
  "res": ""
}
```

wms.inventory.add 盘点单添加

- 系统级参数
- 业务级参数
- items
- 响应结果

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	wms.inventory.add	方法
node_id	String	是	3995507852	绑定状态
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	是否必须	示例	描述
inventory_bn	String	是	201507021313004060	盘点单号
warehouse	String	否	wsm_a1	仓库编号
remark	String	否	这是备注	备注
operate_time	String	否	2015-07-03 10:10:10	操作时间
item	ITEM	是		明细 需json_encode

items

名称	类型	是否必须	示例	描述
product_bn	String	是	sp123	货号
normal_num	int	是	1	良品
defective_num	int	是	1	不良品

响应结果

json格式:

```
{  
  "rsp": "succ",  
  "data": "",  
  "res": ""  
}
```

wms.stock.quantity 库存对账

- [系统级参数](#)
- [业务级参数](#)
- [items](#)
- [响应结果](#)

系统级参数

名称	类型	是否必须	示例	描述
v	String	否	1.0	版本
format	String	否	json	格式, 支持: json、xml
sign	String	是	0d72e9537ece8864b7e29d392509fda7	签名
method	String	是	wms.stock.quantity	方法
node_id	String	是	3995507852	绑定状态
timestamp	int	是	1435912385	时间戳

业务级参数

名称	类型	是否必须	示例	描述
operate_time	String	否	2015-07-03 10:10:10	操作时间
item	ITEM	是		明细 需json_encode

items

名称	类型	是否必须	示例	描述
warehouse	String	是	sp123	仓储
logistics	int	是	1	物流编码
product_bn	String	是	sp123	货号
normal_num	int	是	1	良品
defective_num	int	是	1	不良品

响应结果

json格式:

```
{
  "rsp": "succ",
  "data": "",
  "res": ""
}
```

注：错误的返回格式：

```
{
  "rsp": "fail",
  "data": "",
  "res": "",
  "err_msg": ""
}
```