

电商大师 ERP API 接口说明

v1.1

文档编号：BCSH_P1007_D05V1

文档控制

更改记录

日期	作者	版本	更改参考
2014-05-12	Henry	V1.0	创建
2014-06-04		V1.1	加入售后接口

查阅

姓名	职位	签字

目 录

文档控制	2
第一章 前言	12
1.1 业务背景	12
1.2 授权说明	12
第二章 核心接口	14
2.1 店铺查询	14
2.1.1 应用场景	14
2.1.2 规则	14
2.1.3 API 内容	14
2.2 仓库查询	17
2.2.1 应用场景	17
2.2.2 规则	17
2.2.3 API 内容	17
2.3 快递公司查询	19
2.3.1 应用场景	19
2.3.2 规则	19
2.3.3 API 内容	19

2.4 商品查询.....	21
2.4.1 应用场景.....	21
2.4.2 规则.....	21
2.4.3 API 内容	21
2.5 订单查询.....	26
2.5.1 应用场景.....	26
2.5.2 规则.....	26
2.5.3 API 内容	26
2.6 订单发货.....	35
2.6.1 应用场景.....	35
2.6.2 规则.....	35
2.6.3 API 内容	35
2.7 库存更新.....	37
2.7.1 应用场景.....	37
2.7.2 规则.....	37
2.7.3 API 内容	37
2.8 售后单查询.....	39
2.8.1 应用场景.....	39
2.8.2 规则.....	39
2.8.3 API 内容	39
2.9 售后单入库.....	47

2.9.1 应用场景.....	47
2.9.2 规则.....	47
2.9.3 API 内容	47
2.10 供应商查询	48
2.10.1 应用场景	48
2.10.2 规则	48
2.10.3 API 内容	48
2.11 采购订单查询	50
2.11.1 应用场景	50
2.11.2 规则	50
2.11.3 API 内容	50
2.12 采购订单收货	56
2.12.1 应用场景	56
2.12.2 规则	56
2.12.3 API 内容	56
2.13 采购退货草稿查询	57
2.13.1 应用场景	57
2.13.2 规则	58
2.13.3 API 内容	58
2.14 采购退货	63
2.14.1 应用场景	63

2.14.2 规则	63
2.14.3 API 内容	63
2.15 宝贝查询	64
2.15.1 应用场景	64
2.15.2 规则	65
2.15.3 API 内容	65
2.16 订单拆单	71
2.16.1 应用场景	71
2.16.2 规则	71
2.16.3 API 内容	71
2.17 售后单更新	72
2.17.1 应用场景	72
2.17.2 规则	72
2.17.3 API 内容	72
2.18 售后单创建	73
2.18.1 应用场景	73
2.18.2 规则	73
2.18.3 API 内容	73
2.19 售后原因查询	75
2.19.1 应用场景	75
2.19.2 规则	75

2.19.3 API 内容	76
2.20 商品库存查询	77
2.20.1 应用场景	77
2.20.2 规则	77
2.20.3 API 内容	77
2.21 订单创建	79
2.21.1 应用场景	79
2.21.2 规则	79
2.21.3 API 内容	79
2.22 商品创建	81
2.22.1 应用场景	81
2.22.2 规则	81
2.22.3 API 内容	82
2.23 组合商品创建	84
2.23.1 应用场景	84
2.23.2 规则	84
2.23.3 API 内容	84
2.24 疑难件创建	85
2.24.1 应用场景	85
2.24.2 规则	85
2.24.3 API 内容	85

2.25 订单更新	86
2.25.1 应用场景	86
2.25.2 规则	86
2.25.3 API 内容	86
2.26 账单查询	88
2.26.1 应用场景	88
2.26.2 规则	88
2.26.3 API 内容	88
2.27 商品类目查询	90
2.27.1 应用场景	90
2.27.2 规则	91
2.27.3 API 内容	91
2.28 组合商品查询	93
2.28.1 应用场景	93
2.28.2 规则	93
2.28.3 API 内容	93
2.29 平台订单推送	96
2.29.1 应用场景	96
2.29.2 规则	96
2.29.3 API 内容	96
2.30 客户创建	99

2.30.1 应用场景	99
2.30.2 规则	99
2.30.3 API 内容	100
2.31 宝贝推送	101
2.31.1 应用场景	101
2.31.2 规则	101
2.31.3 API 内容	101
2.32 转储单查询	103
2.32.1 应用场景	103
2.32.2 规则	103
2.32.3 API 内容	103
2.33 转储单行更新	107
2.33.1 应用场景	107
2.33.2 规则	107
2.33.3 API 内容	107
2.34 订单取消	108
2.34.1 应用场景	108
2.34.2 规则	108
2.34.3 API 内容	108
2.35 支付方式查询	109
2.35.1 应用场景	109

2.35.2 规则	109
2.35.3 API 内容	109
2.36 杂项出入库类型查询	111
2.36.1 应用场景	111
2.36.2 规则	111
2.36.3 API 内容	111
2.37 杂项出入库创建	113
2.37.1 应用场景	113
2.37.2 规则	113
2.37.3 API 内容	113
2.38 订单配货	114
2.38.1 应用场景	114
2.38.2 规则	114
2.38.3 API 内容	114
2.39 订单打回待审	115
2.39.1 应用场景	115
2.39.2 规则	115
2.39.3 API 内容	115
2.40 采购订单创建	116
2.40.1 应用场景	116
2.40.2 规则	116

2.40.3 API 内容	116
附录：对应关系	117

第一章 前言

1.1 业务背景

本档为第三方 ERP、WMS 与电商大师对接技术实现方案。其中主要内容为技术接口说明，包括了系统主数据、商品、订单、库存相关接口技术说明。

电商大师 ERP 的 API 接口为 REST 模式，附带了.NET 实现范例。

1.2 授权说明

当前 API 授权仅需在 HTTP REQUEST 头中加入相关登陆信息即可。

每次请求头中必须包含以下五个参数：

Company： 公司名称

LoginName： 登录名（需要在系统中新建用户，不建议使用管理员账户）

Password： 登陆密码

Method： 请求 API 方法名

Version： 版本默认 1.0

Sign: 签名，当 ERP 基础设置中启用了 API 签名 会对签名进行验证。XW

(签名启用： 基础信息 > 基础设置 > 其它设置 > 启用 API 签名)

对 API 密钥（密钥和 ERP 中设置的密钥要一致）和调用参数进行 hmac 加密获得

签名获取方法如下：

根据接口参数名称将所有请求参数按照字母先后顺序排序:key + value key + value

例如：将 foo=1,bar=2,baz=3 排序为 bar=2,baz=3,foo=1，参数名和参数值链接后，

得到拼装字符串 bar2baz3foo1。进行 HMACSHA1 加密。具体可参考 sdk 代码。

注意：Company,LoginName,Password 必须使用 Base64 转码，具体参考 SDK 范例。

LineCustomTax 行税金 如果没有填 0

LineCustomTotal 不含税总价 如果没有填 0

修改 ;

无

第二章 核心接口

2.1 店铺查询

2.1.1 应用场景

查询商家目前在电商大师 ERP 中的所有授权店铺信息。

2.1.2 规则

登陆账户所需系统权限: 基础信息/店铺管理

2.1.3 API 内容

基本信息 :

API 名称	B1EC2.Shop.Query
功能描述	获取商家所有授权店铺

输入参数 : 无

返回结果 :

名称	类型	描述
Shops 店铺对象	Object	面单号集合
└ ShopId, 店铺 ID(int)		ShopType:Taobao,Tmall,Jingdong,YHD,Sunin
└ ShopName, 店铺名称(string,50)		g,Amazon,KDT,Offline
└ ShopType, 店铺类型(string,20)		淘宝 天猫 京东 一号店 苏宁 亚马逊 口袋通 线

		下
--	--	---

JSON 范例：

```
{
  "Shops": [
    {
      "ShopId": 1,
      "ShopName": "Willey 淘宝",
      "ShopType": "Taobao"
    },
    {
      "ShopId": 2,
      "ShopName": "XX 食品旗舰店",
      "ShopType": "Tmall"
    },
    {
      "ShopId": 3,
      "ShopName": "线下店",
      "ShopType": "Offline"
    }
  ]
}
```


2.2 仓库查询

2.2.1 应用场景

查询商家目前在电商大师 ERP 中的所有可用仓库和库区信息。

2.2.2 规则

- 登陆账户所需系统权限: 库存管理/仓库管理
- 电商大师中分为仓库和库区, 因此分为仓库库和库区库存。
 - 仓库库存, 总库存数量和库存成本金额。
 - 库区库存, 进库存数量, 不含库存成本。
- 库区存在设置, 库存是否可用。该属性影响仓库可用库存数量。

2.2.3 API 内容

基本信息:

API 名称	B1EC2.Warehouse.Query
功能描述	获取商家仓库库区信息

输入参数: 无

返回结果:

名称	类型	描述
Warehouses 仓库对象	Object	
└─ WhsId, 仓库 ID(int)		
└─ WhsCode, 仓库代码(string,8)		
└─ WhsName, 仓库名称(string,20)		
└─ WhsArea 库区对象		
└─ WhsAreaId, 库区 ID(int)		

└─ WhsAreaCode, 库区编号(string,10)		
└─ AreaName, 库区名(string,10)		
└─ IsAvailable, 库存是否可用(bool)		

JSON 范例：

```
{
  "Warehouses": [
    {
      "WhsId": 1,
      "WhsCode": "10",
      "WhsName": "主仓库",
      "WhsAreas": [
        {
          "WhsAreaId": 1,
          "WhsAreaCode": "1010",
          "AreaName": "次品区",
          "IsAvailable": true
        },
        {
          "WhsAreaId": 2,
          "WhsAreaCode": "1020",
          "AreaName": "收货区",
          "IsAvailable": true
        },
        {
          "WhsAreaId": 3,
          "WhsAreaCode": "1030",
          "AreaName": "质检区",
          "IsAvailable": true
        },
        {
          "WhsAreaId": 4,
          "WhsAreaCode": "1040",
          "AreaName": "退货区",
          "IsAvailable": true
        }
      ]
    }
  ]
}
```

```

    "WhsAreaId": 5,
    "WhsAreaCode": "1050",
    "AreaName": "整库区",
    "IsAvailable": true
  }
]
}
]
}

```

2.3 快递公司查询

2.3.1 应用场景

查询商家目前在电商大师 ERP 中的所有可用快递公司信息。

2.3.2 规则

- 登陆账户所需系统权限: 基础信息/物流管理
- 快递公司编码为淘宝快递编码, (注意:编码可能重复)
- 订单发货回传需要使用快递 ID 而不是编码

2.3.3 API 内容

基本信息：

API 名称	B1EC2.Express.Query
功能描述	获取商家快递公司信息

输入参数：无

返回结果：

名称	类型	描述
Express 快递对象	Object	
└ ExpressId, 快递 ID(int)		

└ ExpressCode, 快递编码(string,8)		
└ ExpressName, 快递名称(string,8)		
└ ShipLevel, 配送优先级(int)		

JSON 范例：

```
{
  "Expresses": [
    {
      "ExpressId": 1,
      "ExpressCode": "STO",
      "ExpressName": "申通 E 物流",
      "ShipLevel": 2
    },
    {
      "ExpressId": 2,
      "ExpressCode": "YTO",
      "ExpressName": "圆通速递",
      "ShipLevel": 3
    },
    {
      "ExpressId": 3,
      "ExpressCode": "YUNDA",
      "ExpressName": "韵达快运",
      "ShipLevel": 4
    },
    {
      "ExpressId": 4,
      "ExpressCode": "SF",
      "ExpressName": "顺丰速运",
      "ShipLevel": 5
    },
    {
      "ExpressId": 5,
      "ExpressCode": "SF",
      "ExpressName": "顺丰港澳台",
      "ShipLevel": 6
    }
  ]
}
```

2.4 商品查询

2.4.1 应用场景

查询电商大师中的定义的商品主数据。

2.4.2 规则

- 登陆账户所需系统权限: 商品管理/商品资料
- 查询必须提供商品修改开始时间和截至时间参数。
- 商品主数据分为两种, 无规格商品、有规格商品
 - 对于无规格商品, 系统库存中的 SKU CODE 为商品货号 ITEM CODE。
 - 对于有规格商品, 系统库存中的 SKU CODE 为规格明细中的 SKU CODE。
 - 对于虚拟商品, 系统中无库存数据。
- 不提供组合商品接口, 商家在电商大师中定义组合商品后, 订单转换时会自动将组合商品拆分为明细发货商品。

2.4.3 API 内容

基本信息：

API 名称	B1EC2.ItemInfo.Query
功能描述	获取商家所有商品主数据信息

输入参数：无

名称	类型	是否必须	描述
StartTime, 商品修改开始时间	String	Yes	格式为: 2012-05-12 16:58:00
EndTime, 商品修改结束时间	String	Yes	格式同上
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int33	No	取值范围 : 1~100, 默认值 : 50
Property1 自定义属性 1	String	No	

Property2 自定义属性 2	String	No	
SearchText 商品查询条件	String	No	(商品名称, 商品货号)
VirtualStatus 虚拟状态	String	No	T :查询虚拟商品 F :查询非虚拟商品 null 或者 "" 查询所有商品

返回结果：

名称	类型	描述
TotalResults, 搜索到的商品总数	Int32	无
ItemInfos 商品对象	Object	商品集合
└ ItemId, 商品 ID(int)		
└ ItemCode, 货号(string,50)		
└ ItemName, 商品名称(string,100)		
└ PictureUrl, 主图 Url(string,254)		
└ BarCode, 条形码(string,100)		
└ CatCode, 商品类目(int)		
└ CatName, 商品类目(string,50)		
└ CatName1, 一级类目名称(string,50)		
└ CatName2, 二级类目名称(string,50)		
└ CatPath, 类目路径(string,100) 路径范例(包含自身): \10\1010\101010		
└ PurchasePrice, 采购价格(double)		
└ SalesPrice, 销售价格(double)		
└ LowestPrice, 最低售价(double)		
└ MarketPrice, 吊牌价格(double)		
└ Unit, 单位(string,20)		
└ InvoiceName, 开票名称(string,50)		

<ul style="list-style-type: none"> └ Size, 体积(立方米)(double) └ Weight, 重量 (千克) (double) └ BrandId, 品牌 ID(int) └ Brand, 品牌 <ul style="list-style-type: none"> └ BrandId, 品牌 ID(int) └ BrandName, 品牌名称(string,20) └ IsAgent, 是否代销(bool) └ IsVirtual, 是否虚拟商品(bool) └ IsStopProducte, 是否停产(bool) └ IsSerialNumber, 是否序列号管理(bool) └ Memo, 备注(string,254) └ CreatedTime, 创建日期(DateTime) └ UpdateTime, 最后修改日期(DateTime) └ UserSign, 修改人(string) └ Property1, 自定义属性 1(string,100) └ Property2, 自定义属性 2(string,100) └ Property3, 自定义属性 3(string,100) └ Property4, 自定义属性 4(string,100) └ Property5, 自定义属性 5(string,100) └ Property6, 自定义属性 6(string,100) └ Property7, 自定义属性 7(string,100) └ Property8, 自定义属性 8(string,100) └ Property9, 自定义属性 9(string,100) └ Property10, 自定义属性 10(string,100) └ Property11, 自定义属性 11(string,100) └ Property12, 自定义属性 12(string,100) └ Lines, SKU 行(list) 规格对象 <ul style="list-style-type: none"> └ SkuId, SKU ID(int) 		
---	--	--

<ul style="list-style-type: none"> └ SkuCode, 规格编码(string,50) └ SkuName, 规格名称(string,50) └ BarCode, 条形码(string,50) └ Property1, 销售属性 1(string,100) └ Property2, 销售属性 2(string,100) └ PurchasePrice, 采购价格(double) └ SalesPrice, 销售价格(double) └ LowestPrice, 最低售价(double) └ MarketPrice, 吊牌价格(double) └ Size, 体积(立方米)(double) └ Weight, 重量 (千克) (double) 		
---	--	--

JSON 范例 :

```
{
  "TotalResults": 2,
  "ItemInfos": [
    {
      "Brand": null,
      "Lines": [
        {
          "SkuId": 1,
          "SkuCode": "test1231",
          "SkuName": "属性 1 属性 2",
          "BarCode": "",
          "Property1": "属性 1",
          "Property2": "属性 2",
          "PurchasePrice": 0,
          "SalesPrice": 0,
          "LowestPrice": 0,
          "MarketPrice": 0,
          "Size": 0,
          "Weight": 0
        }
      ]
    },
    {
      "ItemId": 6,
```



```
"ItemCode": "test123",
"ItemName": "测试 123",
"PictureUrl": "",
"BarCode": "",
"CatCode": 0,
"CatName": "",
"CatName1": "",
"CatName2": "",
"CatPath": "",
"PurchasePrice": 0,
"SalesPrice": 30,
"LowestPrice": 0,
"MarketPrice": 0,
"Unit": "",
"InvoiceName": "123",
"Size": 0,
"Weight": 0,
"BrandId": null,
"IsAgent": false,
"IsVirtual": true,
"IsStopProducte": false,
"IsSerialNumber": false,
"Memo": "test",
"CreatedTime": "2014-05-12T11:39:52",
"UpdateTime": "2014-05-12T14:56:58",
"UserSign": "演示系统\admin",
"Property1": "",
"Property2": "",
"Property3": "",
"Property4": "",
"Property5": "",
"Property6": "",
"Property7": "",
"Property8": "",
"Property9": "",
"Property10": "",
"Property11": "",
"Property12": ""
}
]
}
```

2.5 订单查询

2.5.1 应用场景

根据开始修改时间和截至修改时间来查询系统中订单数据。

2.5.2 规则

- 登陆账户所需系统权限: 销售管理/订单查询
- 查询必须提供订单修改开始时间和截至时间参数。
- **必须采用倒序的分页方式（从最后一页往回取）才能避免漏单问题。**

2.5.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.Query
功能描述	获取商家所有订单数据

输入参数：

名称	类型	是否必须	描述
WhsId, 仓库 ID	Int32	No	默认查询所有仓库的数据，除了默认值外每次只能查询一个仓库

ShopId 店铺 Id	Int32	No	默认查询所有店铺的数据，除了默认值外每次只能查询一个店铺
OrderNo 订单号	String	No	按系统订单号查询某个订单
TimeType 时间类型	Int32	No	默认为 10 10：按更新时间；20：按创建时间 30：按照付款时间; 40: 按发货时间
Status, 交易状态	Int32	No	默认查询所有交易状态的数据，除了默认值外每次只能查询一种状态。 10: 待客审;20: 待财审;30: 审批通过;40: 开始拣货;43: 已验货;45: 已称重;60: 已交货;70: 已完成;80: 已关闭;90: 已取消;99: 已删除;
StartTime, 订单修改开始时间	String	Yes	格式为: 2012-05-12 16:58:00
EndTime, 订单修改结束时间	String	Yes	格式同上
PageNum, 页码	Int32	No	取值范围: 大于零的整数; 默认值: 1。
PageSize, 每页条数	Int32	No	取值范围: 1~100, 默认值: 50
UserDefinedField1 自定义字段 1	String	No	
UserDefinedField2 自定义字段 2	String	No	
SerialNumber 序列号	String	No	订单行包含此序列号
ShopOrderNo, 平台订单编号	String	No	
InvoiceNo 发票号码	String	No	

返回结果：

名称	类型	描述
TotalResults, 搜索到的订单总数	Int32	订单集合
SalesOrder 销售订单主表	Object	SalesOrder 销售订单主表
└─ OrderId, 订单 ID(int)		1) 订单类型 OrderType : (10: 正常订单; 20: 货到付款; 30: 虚拟订单; 40: 代销订单)
└─ Version, 版本(int)		

<ul style="list-style-type: none"> └ OrderNo, 系统订单号(string) └ ShopId, 店铺 ID(int) └ Shop, 店铺(Shop) 注：只返回店铺名称 <ul style="list-style-type: none"> └ ShopId, 店铺 ID(int) └ ShopName, 店铺名称(string,50) └ ShopType, 店铺类型(string,20) └ CustomerCode, 会员代码(string) └ CustomerName, 会员名称(string) └ WhsId, 仓库 ID(int) └ Warehouse, 仓库(Warehouse) 注：只返回仓库名称 <ul style="list-style-type: none"> └ WhsId, 仓库 ID(int) └ WhsCode, 仓库代码(string,8) └ WhsName, 仓库名称(string,20) └ OrderType, 订单类型(int) 注：请看描述 └ OrderStatus, 系统订单状态(int) 注：请看描述 └ OrderTag, 订单标签(int) 注：请看描述 └ ReasonId, 原因 ID(int) └ HoldingReason, 拦截原因 <ul style="list-style-type: none"> └ ReasonId, 原因 ID(int) └ ReasonName, 原因名称(string) └ HoldingMemo, 拦截备注(string,254) └ TotalNum, 总件数(double) └ DiscountFee, 折扣金额(double) └ AdjustFee, 调整金额(double) └ GoodsTotal, 商品总额(double) └ OrderTotal, 应付金额(double) └ CalcTotal, 计算金额(double) 注：请看描述 └ StockTotal, 成本总计(double) 	<p>2) 系统订单状态 OrderStatus : 10:待客审;20:待财审;30:审批通过;40:开始拣货;43:已验货;45:已称重;60:已交货;70:已完成;80:已取消;</p> <p>3) 订单标签 OrderTag :使用组合码体现, 如 1 表示合单,2 表示拆单,4 表示急单; 3 表示合单+拆单,5 表示合单+急单,7 表示合单+拆单+急单; 1,合;2,拆;4,聚;8,急;16,退;32,换;64,手;128,分;256,预;512,超;1024,途;2048,补;</p> <p>4) 计算金额 CalcTotal : 按此金额计算是否进入财审;金额具体计算规则参考设置中 ApprovalOrder2Type 字段</p> <p>5) 平台订单状态 GoodsTotal : 平台订单状态描述, 如已付款, 等待发货等等。都为中文描述, 仅备注作用; 每次订单下载时, 更新本字段</p> <p>6) 打印标签 PrintTag :使用组合码体现, 范例参考订单标签; 1,打印快递单;2,打印交货单;4,打印拣货单;</p> <p>7) 最后修改类型 LastModifiedType : 10,制单;12,合单;14,拆单;20,客审;21 财审;22,打回待审;23 客审进财审;24,配货;25,验货;26,称重;27,交货;28,交易完成;29,关闭订单;30,取消;40,拦截;42 解除拦截;50,打印物流;52,打印发货;54,打印拣货;60,更新;</p> <p>8) 订单拦截原因: 1, 商品不存在; 2, 订单缺货; 3, 退款中; 4, 发货回写异常; 5, 系统预留。</p>
--	---

<ul style="list-style-type: none"> └ ShopOrderNo, 平台订单编号(string,254) 注：网店订单编号，如果合单则多个订单通过逗号分隔 └ ShopOrderStatus, 平台订单状态(string,10) 注：请看描述 └ ShopCreatedTime, 网站下单时间(DateTime) └ ShopPayTime, 网站付款时间(DateTime) └ BuyerMemo, 买家留言(string,1000) └ SellerMemo, 卖家备注(string,1000) └ OrderMemo, 系统备注(string,1000) └ ShopFlag, 网店旗帜(string,6) └ InvoiceStatus, 开票状态(int) 注：10:未开票;20:已开票; └ InvoiceType, 开票类型(int) 注：10:普票;20:增票; └ InvoiceTotal, 开票金额(double) └ InvoiceName, 发票抬头(string,100) └ InvoiceMemo, 开票备注(string,254) └ InvoiceExpressNo, 发票快递号(string,20) └ BuyerAlipayNo, 买家支付宝(string,100) └ ReceiverName, 收货人姓名(string,50) └ ReceiverState, 收货人省份(string,32) └ ReceiverCity, 收货人城市(string,32) └ ReceiverDistrict, 收货人地区(string,32) └ ReceiverAddress, 收货人地址(string,228) └ ReceiverZip, 收货人邮编(string,6) └ ReceiverMobile, 收货人手机(string,30) └ ReceiverPhone, 收货人电话(string,30) └ ExpressId, 快递公司 ID(int) └ Express, 快递公司(Express) 注：只返回快递名称 		
---	--	--

<ul style="list-style-type: none"> └ ExpressId, 快递公司 ID(int) └ ExpressCode, 快递编码(string,8) 唯一 └ ExpressName, 快递名称(string) └ ExpressTrackNo, 快递单号(string,20) └ PrintTag, 打印标签(int) 注：请看描述 └ PrintBatchNo, 快递单打印批次(string) └ ExpectedDeliveryDate, 要求发货日(DateTime) └ DeliveryTime, 实际发货时间(DateTime) └ EndTime, 交易完成时间(DateTime) └ StdPostFee, 标准物流成本(double) └ CostPostFee, 称重物流成本(double) └ ActPostFee, 物流费用(double) └ EstimatedWeight, 包裹标准重量(double) └ ActualWeight, 包裹称重重量(double) └ Creator, 创建人(string,20) Not Null └ CreatedTime, 创建时间(DateTime) └ LastModifiedTime, 最后修改时间(DateTime) └ LastModifiedUser, 最后修改人(string,20) └ LastModifiedType, 最后修改类型(int) 注：请看描述 └ UserDefinedField1, 自定义字段 1(string) └ UserDefinedField2, 自定义字段 2(string) └ UserDefinedField3, 自定义字段 3(string) └ UserDefinedField4, 自定义字段 4(string) └ UserDefinedField5, 自定义字段 5(string) └ UserDefinedField6, 自定义字段 6(string) └ UserDefinedField7, 自定义字段 7(string) └ UserDefinedField8, 自定义字段 8(string) └ SalesOrderLine 订单行 		
---	--	--

<ul style="list-style-type: none"> └ OrderId, 订单 ID(int) 注: 虚拟商品不放入总件数中 └ LineNum, 行号(int) └ Version, 日志版本(int) └ LineStatus, 行状态(int) 注 : 10:未清;20:已清;30:已取消; └ ShopOrderNo, 平台订单编号(string,20) └ ShopLineNo, 平台订单行编号(string,20) └ ShopLineStatus, 平台订单行状态(string,10) └ ShopItemName, 平台商品名称(string,100) └ ItemId, 商品 ID(int) └ ItemName, 商品名称(string,100) Not Null └ SkuCode, 商品编码(string,50) 注 : 如果为 item 没有 sku,则 skucode 等于货号 └ SkuName, 规格名称(string,100) └ BarCode, 商品条码(string,254) └ Unit, 单位(string,20) └ Price, 单价(double) └ Quantity, 数量(double) Normal int └ LineDiscountFee, 行折扣(double) └ LineAdjustFee, 行调整金额(double) └ LineTotal, 行总计(string) └ StockPrice, 库存成本(double) └ StockValue, 库存金额(double) └ IsVirtual, 是否虚拟(bool) └ IsSerialNumber, 是否序列号管理(bool) └ SerialNumbers, 序列号(string) LongText └ Location, 拣货库位(string,100) 		
--	--	--

<ul style="list-style-type: none"> └ LineUdf1, 自定义行字段 1(string,50) └ LineUdf2, 自定义行字段 2(double) └ LineUdf3, 自定义行字段 3(string,50) └ LineUdf4, 自定义行字段 4(string,254) └ LineMemo, 行备注(string) └ Payments, 支付方式 <ul style="list-style-type: none"> └ OrderId 订单 ID └ LineNum 行号 └ PaymentId 付款方式 └ PaymentTotal 付款金额 └ PaymentNo 支付宝付款编号 └ UserDefinedField1 自定义字段 		
---	--	--

JSON 范例 :

```
{
  "TotalResults": 1,
  "SalesOrders": [
    {
      "Shop": {
        "ShopId": 0,
        "ShopName": "Willey 淘宝",
        "ShopType": "Taobao"
      },
      "Warehouse": {
        "WhsId": 0,
        "WhsCode": null,
        "WhsName": "11111111"
      },
      "HoldingReason": null,
      "Express": {
        "ExpressId": 0,
        "ExpressCode": null,
        "ExpressName": "圆通速递"
      },
      "Lines": [
```



```
{
  "OrderId": 20,
  "LineNum": 0,
  "Version": 5,
  "LineStatus": 20,
  "ShopOrderNo": "471403169149807",
  "ShopLineNo": "471403169159807",
  "ShopLineStatus": "卖家已付款",
  "ShopItemName": "牛肉礼盒 1000g 牛肉礼盒装包邮",
  "ItemId": 3,
  "ItemName": "5040185",
  "SkuCode": "5040185",
  "SkuName": null,
  "BarCode": "",
  "Unit": "",
  "Price": 198,
  "Quantity": 1,
  "LineDiscountFee": 85.1,
  "LineAdjustFee": 0,
  "LineTotal": 112.9,
  "StockPrice": 0,
  "StockValue": 0,
  "IsVirtual": false,
  "IsSerialNumber": false,
  "SerialNumbers": null,
  "Location": null,
  "LineUdf1": null,
  "LineUdf2": null,
  "LineUdf3": null,
  "LineUdf4": null,
  "LineMemo": null,
}
],
"Payments": [
  {
    "OrderId": 20,
    "LineNum": 0,
    "PaymentId": 7,
    "PaymentTotal": 107.3,
    "PaymentNo": "",
    "UserDefinedField1": "16012816181666059967"
  }
],
```

```
"OrderId": 20,  
"Version": 5,  
"OrderNo": "20140504000020",  
"ShopId": 1,  
"CustomerCode": "100002",  
"CustomerName": "火乐**",  
"WhsId": 2,  
"OrderType": 10,  
"OrderStatus": 60,  
"OrderTag": 0,  
"ReasonId": null,  
"HoldingMemo": null,  
"TotalNum": 2,  
"DiscountFee": 10,  
"AdjustFee": 0,  
"GoodsTotal": 456,  
"OrderTotal": 240.9,  
"CalcTotal": 0,  
"StockTotal": 0,  
"ShopOrderNo": "471403169149807",  
"ShopOrderStatus": "交易成功",  
"ShopCreatedTime": "2014-02-20T09:06:14",  
"ShopPayTime": "2014-04-20T09:14:42",  
"BuyerMemo": null,  
"SellerMemo": null,  
"OrderMemo": null,  
"ShopFlag": null,  
"InvoiceStatus": 10,  
"InvoiceType": 10,  
"InvoiceTotal": 240.9,  
"InvoiceName": null,  
"InvoiceMemo": null,  
"InvoiceExpressNo": null,  
"BuyerAlipayNo": null,  
"ReceiverName": "张*",  
"ReceiverState": "四川省",  
"ReceiverCity": "绵阳市",  
"ReceiverDistrict": "涪城区",  
"ReceiverAddress": "****大学北四 A-121",  
"ReceiverZip": "621000",  
"ReceiverMobile": "1588288****",  
"ReceiverPhone": "",  
"ExpressId": 2,
```

```
"ExpressTrackNo": "1234554321",
"PrintTag": 0,
"PrintBatchNo": null,
"ExpectedDeliveryDate": null,
"DeliveryTime": "2014-05-12T14:31:15",
"EndTime": "2014-02-24T13:29:07",
"StdPostFee": 0,
"CostPostFee": 0,
"ActPostFee": 0,
"EstimatedWeight": 0,
"ActualWeight": 0,
"Creator": "admin",
"CreatedTime": "2014-05-04T15:54:22",
"LastModifiedTime": "2014-05-12T14:31:15",
"LastModifiedUser": "演示系统\\admin",
"LastModifiedType": 60,
"UserDefinedField1": null
}
]
}
```

2.6 订单发货

2.6.1 应用场景

根据订单 ID、快递 ID、快递单号更新订单发货状态。

2.6.2 规则

- 登陆账户所需系统权限: 销售管理/订单查询/订单发货
- 如果不提供快递 ID, 回写平台时候认为无需物流发货。

2.6.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.Deliver
--------	--------------------------

功能描述	更新订单发货状态以及快递公司信息
------	------------------

输入参数：

名称	类型	是否必须	描述
OrderId, 订单 ID	Int32	No	OrderId 与 OrderNo 必须二选一
OrderNo, 订单号	String	No	OrderId 与 OrderNo 必须二选一
ExpressId, 物流公司 ID	Int32	No	
ExpressTrackNo, 快递单号	Int32	No	
Weight, 包裹重量	Double	No	
SerialNumberList 序列号	String	No	序列号 包含：订单 LineNum:序列号; 范例：0:XXXXXX;1:XXXXXX;
InvoiceNo	String	No	
InvoiceDate	DateTime	No	开票日期
OrderMemo	String	No	
ReasonId ,拦截原因 Id	Int32	No	ReasonId 设置为 0,则取消拦截
HoldingMemo, 拦截备注	String	No	
UserDefinedField1	String	No	
UserDefinedField2	String	No	
UserDefinedField3	String	No	
UserDefinedField4	String	No	
UserDefinedField5	String	No	
UserDefinedField6	double	No	
UserDefinedField7	double	No	
UserDefinedField8	DateTime	No	

返回结果：

如果发货成功，则返回空 Json。否则返回错误码和错误信息。

2.7 库存更新

2.7.1 应用场景

根据库位编码，SKU 编码，数量，成本价更新系统库存信息。

电商大师系统中是区分仓库和库区的，如果第三方产品无库区，可以将系统中库区编码设置为本地系统的仓库编码即可。

一次请求中，如果任一条库存更新错误，则请求中的库存更新全部回滚。

2.7.2 规则

- 登陆账户所需系统权限: 库存管理/库存盘点
- 一次请求最多只能更新 100 个库存，如果超过，则请分批次更新。
- 如果不希望更新库存成本,则成本字段设置为 0;

2.7.3 API 内容

基本信息：

API 名称	B1EC2.Stock.Update
功能描述	更新系统库存信息

输入参数：无

名称	类型	是否必须	描述
Stock, 库存信息	String	Yes	库存信息,包含 (库区编码:商品编码:数量:成本;) 范例如下: W01:1201:10:12.5;W01:1202:5:0;

Confirmed	Bool	No	默认为 True
-----------	------	----	----------

返回结果：

如果更新成功，则返回空 Json。否则返回错误码和错误信息。

2.8 售后单查询

2.8.1 应用场景

根据开始修改时间和截至修改时间来查询系统中售后单数据。

2.8.2 规则

- 登陆账户所需系统权限: 售后管理
- 查询必须提供售后单修改开始时间和截至时间参数。
- 必须采用倒序的分页方式（从最后一页往回取）才能避免漏单问题。

2.8.3 API 内容

基本信息：

API 名称	B1EC2.AfterSalesService.Query
功能描述	获取商家所有售后单数据

输入参数：

名称	类型	是否必须	描述
Type, 售后类型	Int32	No	默认查询所有售后类型的数据，除了默认值外每次只能查询一个类型 10：退货退款；20：仅退款；30：换货；40：疑难件；50:补发货；60：维修；
Status, 售后状态	Int32	No	默认查询所有交易状态的数据，除了默认值外每次只能查询一种状

			态。 10:草稿;20:待处理;30:已完结 ; 90: 删除 ;
AfterSaleServiceNo , 售后系统单号	String	No	
ReceiverName,收货人姓名	String	No	
SerialNumber	String	No	原订单包含此序列号
UserDefinedField1	String	No	
StartTime, 售后单修改开始时间	String	Yes	格式为: 2012-05-12 16:58:00
EndTime, 售后单修改结束时间	String	Yes	格式同上
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int32	No	取值范围 : 1~100 , 默认值 : 50

返回结果 :

AfterSaleService 售后订单主表

- └ Shop, 店铺(Shop) 注：只返回店铺名称
 - └ ShopId, 店铺 ID(int)
 - └ ShopName, 店铺名称(string,50)
 - └ ShopType, 店铺类型(string,20)
- └ ReturnWarehouse, 退货仓库(Warehouse) 注：只返回仓库名称
 - └ WhsId, 仓库 ID(int)
 - └ WhsCode, 仓库代码(string,8)
 - └ WhsName, 仓库名称(string,20)
- └ ReturnExpress, 退货快递公司(Express)
 - └ ExpressId, 快递公司 ID(int)
 - └ ExpressCode, 快递编码(string,8) 唯一
 - └ ExpressName, 快递名称(string)
- └ AfterSaleServiceId, 售后单 ID(int)
- └ Version, 版本(int)
- └ AfterSaleServiceNo, 流水号(string)注：40+年月日+六位流水号
- └ ShopId, 店铺 ID(int)
- └ CustomerCode, 客户代码(string,10)
- └ CustomerName, 客户名称(string,50)
- └ Type, 售后类型(int) 注：退货退款 10 仅退款 20 换货 30 疑难 40 补发货 50
- └ Status, 售后状态(int) 注：草稿 10 待处理 20 已完结 30 删除 90
- └ ReturnWhsId, 退货仓库(int)
- └ OrderId, 订单 ID(int)
- └ SalesOrder, 原订单 SalesOrder
 - └ OrderID, 原订单 ID (int)
 - └ OrderNo, 原订单号 (string,14)
 - └ OrderStatus, 原订单状态 (int)
- └ OrderNo, 系统订单号(string,14)

- └ ShopOrderStatus, 网站订单状态(string,10)
- └ ExchangeOrderId, 换货新订单 ID(int)
- └ ExchangeOrder, 换货新订单 SalesOrder
 - └ OrderID, 换货新订单 ID (int)
 - └ OrderNo, 换货新订单号 (string,14)
 - └ OrderStatus, 换货新订单状态 (int)
- └ ShopRefundNo, 网站售后单号(string,30)
- └ ReasonId, 退回原因(int)
- └ RefundMemo, 退回备注(string,254)
- └ RefundFeeTotal, 应退金额(double)
- └ ActualRefundFee, 实退金额(double)
- └ ActualFillUpFee, 实补金额(double)
- └ FillUpFee, 应补金额(double)
- └ GoodsTotal, 商品应退总计(double)
- └ ReceiverName, 收货人姓名(string,50)
- └ ReceiverState, 收货人省份(string,32)
- └ ReceiverCity, 收货人城市(string,32)
- └ ReceiverDistrict, 收货人地区(string,32)
- └ ReceiverAddress, 收货人地址(string,228)
- └ ReceiverZip, 收货人邮编(string,6)
- └ ReceiverMobile, 收货人手机(string,30)
- └ ReceiverPhone, 收货人电话(string,30)
- └ SellerMemo, 卖家备注(string,1000)
- └ DeliverExpressId, 发货快递公司 ID(int) 注 : 换货/补发货 才有
- └ ReturnExpressId, 退回快递公司 ID(int)
- └ ExpressTrackNo, 退货快递单号(string,20)
- └ BuyerAlipayNo, 买家支付宝(string,100)
- └ CreatedTime, 创建时间(DateTime)

- └ UpdateTime, 更新时间(DateTime)
- └ LastModifiedUser, 最后修改人(string,20)
- └ LastModifiedType, 最后修改类型(int) 注 : 10,制单;21,入库;22,确认;30,打款 ; 60,更新 ; 90 , 删除;
- └ AfterSaleItemLine 售后订单行
 - └ ItemLines 售后单行
 - └ AfterSaleServiceId, 售后单 ID(int)
 - └ LineNum, 行号(int)
 - └ Version, 版本号(int)
 - └ SkuCode, Sku(string,50)
 - └ ItemId, 商品 ID(int)
 - └ ShopOrderNo, 平台订单编号(int)
 - └ ShopLineNo, 平台订单行编号(int)
 - └ ItemName, 商品名称(string,100)
 - └ SkuName, 型号(string,100)
 - └ Quantity, 商品数量(double)
 - └ Price, 价格(double)
 - └ Unit, 单位(string,20)
 - └ LineTotal, 行总计(double)
 - └ StockPrice, 库存成本(double)
 - └ StockValue, 库存金额(double)
 - └ NeedSerial, 是否序列号管理(bool)
 - └ SerialNumbers, 序列号(string)
- └ SalesOrderSN, 原订单序列号(string)

JSON 范例 :

```
{  
    "TotalResults":1,
```

```
"AfterSaleServices":[
  {
    "Shop":{
      "ShopId":0,
      "ShopName":"京东演示店铺", },
    "ReturnWarehouse":{
      "WhsId":1,
      "WhsName":"主仓库"},
    "ReturnExpress": {
      "ExpressId": 0,
      "ExpressCode": null,
      "ExpressName": null
    },
    "AfterSaleServiceId":14,
    "Version":2,
    "AfterSaleServiceNo":"40140604000014",
    "ShopId":12,
    "CustomerCode":"109444",
    "CustomerName":"18652896152_p",
    "Type":10,
    "Status":20,
    "ReturnWhsId":1,
    "OrderId":15008,
    "OrderNo":"20140527015008",
    "ShopOrderStatus":"完成",
    "ExchangeOrderId":null,
    "ShopRefundNo":null,
    "ReasonId":null,"
    "RefundMemo":"",
    "RefundFeeTotal":7.44,
```

```
"ActualRefundFee":0.0,
"ActualFillUpFee":0.0,
"FillUpFee":0.0,
"GoodsTotal":9900.0,
"ReceiverName":"陈俊",
"ReceiverState":"江苏",
"ReceiverCity":"镇江市",
"ReceiverDistrict":"润州区",
"ReceiverAddress":"江苏镇江市润州区宝塔路街道华都名城荣锦苑 41 幢 103",
"ReceiverZip":"",
"ReceiverMobile":"18652896152",
"ReceiverPhone":"18652896152",
"SellerMemo":"5.28 日快递 男宝宝 红色马儿叮当 N2 ( 1 卤蛋+1 小天使+2 臻爱喜饼 +2 喜
果子+2 海派+4 旺仔 ) 10*7.99 已打单 ( 陆 ) ( 6527316142 ) ",
"DeliverExpressId":2,
"ReturnExpressId":null,
"ExpressTrackNo":"",
"BuyerAlipayNo":"",
"CreatedTime":"2014-06-04T13:13:53",
"UpdateTime":"2014-06-04T13:14:01",
"LastModifiedUser":"演示系统\\admin",
"LastModifiedType":21,
"ItemLines":[
{
"AfterSaleServiceId":14,
"LineNum":0,
"Version":2,
"SkuCode":"13PMX2208850115",
```

```
"ItemId":4,  
"ShopOrderNo":"1460466490",  
"ShopLineNo":"0",  
"ItemName":"韩国犀牛 PAT 商务休闲纯棉透气格子男短袖衬衫 13PMX2208 推荐款",  
"SkuName":"13PMX2208-850 115#/185",  
"Quantity":10.0,  
"Price":990.0,  
"Unit": "",  
"LineTotal":7.44,  
"StockPrice":0.0,  
"StockValue":0.0,  
"NeedSerial":false,  
"SerialNumbers": "",  
"SalesOrderSN": ""}  
],  
}  
]  
}
```

2.9 售后单入库

2.9.1 应用场景

根据售后单 ID 更新售后单入库状态。

2.9.2 规则

- 登陆账户所需系统权限: 售后管理

2.9.3 API 内容

基本信息：

API 名称	B1EC2.AfterSalesService.Receipt
功能描述	更新售后单入库状态

输入参数：无

名称	类型	是否必须	描述
AfterSalesServiceId, 售后单单 ID	Int32	No	Id 与 No 2 选一
AfterSalesServiceNo 售后单单号	String	No	Id 与 No 2 选一

返回结果：

如果发货成功，则返回空 Json。否则返回错误码和错误信息。

2.10 供应商查询

2.10.1 应用场景

查询商家目前在电商大师 ERP 中的所有可用供应商信息。

2.10.2 规则

- 登陆账户所需系统权限: 供应商

2.10.3 API 内容

基本信息：

API 名称	B1EC2.Supplier.Query
功能描述	获取商家供应商信息

输入参数：

名称	类型	是否必须	描述
StartTime, 最后修改开始时间	DateTime	Yes	
EndTime, 最后修改结束时间	DateTime	Yes	

返回结果：

名称	类型	描述
Suppliers 供应商对象 └─ SupplierId, 供应商 ID(int)	Object	

<ul style="list-style-type: none">└ SupplierCode, 供应商代码(string,10)└ SupplierName, 供应商名称(string,20)└ Province , 省份(string,20)└ City , 城市(string,20)└ District, 区(string,20)└ Address , 地址(string,100)		
---	--	--

JSON 范例 :

```
{
  "Suppliers ": [
    {
      "SupplierId ": 1,
      "SupplierCode": "100009",
      "SupplierName": "测试供应商陈磊",
      "Province ": "上海",
      "City": "上海",
      "District ": "上海",
      "Address ": "上海市徐汇区钦州路 10 号",
    }
  ]
}
```

2.11 采购订单查询

2.11.1 应用场景

根据开始修改时间和截至修改时间来查询系统中采购订单数据。

2.11.2 规则

- 登陆账户所需系统权限: 采购管理/采购订单
- 查询必须提供采购订单修改开始时间和截至时间参数。
- **必须采用倒序的分页方式（从最后一页往回取）才能避免漏单问题。**

2.11.3 API 内容

基本信息：

API 名称	B1EC2.PurchaseOrder.Query
功能描述	获取商家所有采购订单数据

输入参数：无

名称	类型	是否必须	描述
WhsId, 仓库 ID	Int32	No	默认查询所有仓库的数据，除了默认值外每次只能查询一个仓库
Status, 交易状态	Int32	No	默认查询所有交易状态的数据，除了默认值外每次只能查询一种状态。

			10:未到货;20:已完成;30:已取消;40:草稿;
UserDefinedField1	String	No	
StartTime, 订单修改开始时间	DateTime	Yes	
EndTime, 订单修改结束时间	DateTime	Yes	
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int32	No	取值范围 : 1~100 , 默认值 : 50

返回结果 :

名称	类型	描述
TotalResults, 搜索到的采购订单总数	Int32	订单集合
PurchaseOrder 采购订单主表	Object	PurchaseOrder 采购订单主表
<ul style="list-style-type: none"> └ OrderId, 订单 ID(int) └ Version, 版本(int) └ OrderNo, 系统订单号(string) └ OrderDate , 业务日期(DateTime) └ CreateTime , 创建时间(DateTime) └ SupplierId , 供应商ID(int) └ WhsId , 仓库ID(int) └ OrderStatus , 订单状态(int) └ TotalQuantity , 总采购量(int) └ TotalCloseQty , 总到货量(int) └ Memo , 备注(string , 254) └ ExpectedReceiptDate , 预计收货日期(DateTime) └ LastModifiedTime , 最后修改日期(DateTime) └ LastModifiedUser , 最后修改人(string , 20) └ CreateUser , 创建人(string , 20) └ UserDefinedField1 自定义字段1(string , 50) └ UserDefinedField2 自定义字段2(string , 50) └ UserDefinedField3 自定义字段3 (string , 100) └ UserDefinedField4 自定义字段4(string , 100) └ UserDefinedField5 自定义字段5(string , 254) └ UserDefinedField6 自定义字段6(double) └ UserDefinedField7 自定义字段7(double) └ UserDefinedField8 自定义字段8(DateTime) 		1) 系统订单状态 OrderStatus : 10:待客 审;20:待财审;30:审批通过;40:开始拣货;43:已 验货;45:已称重;60:已交货;70:已完成;80:已取 消;

<ul style="list-style-type: none"> └ PurchaseOrderLine 订单行 <ul style="list-style-type: none"> └ OrderId, 订单 ID(int) 注: 虚拟商品不放入总件数中 └ LineNum, 行号(int) └ Version, 日志版本(int) └ LineStatus, 行状态(int) 注 :10:未清;20:已清; └ ItemId, 商品 ID(int) └ ItemName, 商品名称(string,100) Not Null └ SkuCode, 商品编码(string,50) 注 : 如果为 item 没有 sku,则 skucode 等于货号 └ SkuName, 规格名称(string,100) └ ItemCode, 商品货号(string,100) └ Price, 单价(double) └ Quantity, 数量(int) └ ReceivedQty, 已入库量(int) └ LineTotal, 行总计(string) └ LineMemo, 行备注(string,254) └ ItemInfoMemo, 商品备注(string , 254) └ Unit, 单位(string , 20) └ Proportion, 比例(int) └ Barcode, 商品条码(string , 254) 		
--	--	--

JSON 范例 :

```
{
  "TotalResults": 1,
  "PurchaseOrders": [
    {
      "Supplier": {
```

```
"SupplierId": 2,
"SupplierCode": "V00001",
"SupplierName": "测试供应商 1",
"Enabled": true,
"Province": "上海",
"City": "上海市辖区",
"District": "卢湾区",
"Address": "钦州路 100 号",
},
"Warehouse": {
  "WhsId": 0,
  "WhsCode": null,
  "WhsName": "主仓库",
},
"Lines": [
  {
    "ItemInfo": {
      "ItemId": 0,
      "ItemCode": "GZ6JBZTT",
      "ItemName": null,
    },
    "OrderId": 5,
    "LineNum": 0,
    "Version": 1,
    "LineStatus": 10,
    "LineStatusEnum": 10,
    "ItemId": 44326,
    "SkuCode": "0816-1",
```

```
"ItemName": "史丹利 12.5MM 系列公制 6 角标准套筒 规格可选 8mm-32mm",
"ItemCode": "GZ6JBZTT",
"SkuName": "24mm ( 88-974-1  ",
"Quantity": 100,
"Price": 88,
"LineTotal": 8800,
"ExpectedReceiptDate": null,
"LineMemo": "",
"ItemInfoMemo": "",
"ReceivedQty": 0,
"ReceiptDate": null,
"Unit": "个",
"Proportion": 1
},
{
  "ItemInfo": {
    "ItemId": 0,
    "ItemCode": "GZ6JBZTT",
    "ItemName": null,
  },
  "OrderId": 5,
  "LineNum": 1,
  "Version": 1,
  "LineStatus": 10,
  "LineStatusEnum": 10,
  "ItemId": 44326,
  "SkuCode": "6059-1",
  "ItemName": "史丹利 12.5MM 系列公制 6 角标准套筒 规格可选 8mm-32mm",
```

```
"ItemCode": "GZ6JBZTT",
"SKUName": "12mm ( 86-109-1-2  ",
"Quantity": 50,
"Price": 66,
"LineTotal": 3300,
"ExpectedReceiptDate": null,
"LineMemo": "",
"ItemInfoMemo": "",
"ReceivedQty": 0,
"ReceiptDate": null,
"Unit": "个",
"Proportion": 1
}
],
"OrderId": 5,
"Version": 1,
"OrderNo": "30140627000005",
"OrderDate": "2014-06-27T00:00:00",
"CreateTime": "2014-06-27T10:14:57",
"SupplierId": 2,
"WhsId": 1,
"OrderStatus": 10,
"OrderStatusEnum": 10,
"TotalQuantity": 150,
"TotalCloseQty": 0,
"Memo": "",
"LastModifiedTime": "2014-06-27T10:14:57",
"LastModifiedUser": "admin",
```

```
"CreateUser": "admin"  
  }  
]  
}
```

2.12 采购订单收货

2.12.1 应用场景

根据采购订单 ID、库区代码、商品行信创建采购收货单

2.12.2 规则

- 登陆账户所需系统权限: 采购收货

2.12.3 API 内容

基本信息：

API 名称	B1EC2.Purchase.Receipt
功能描述	根据采购订单 ID、库区代码、备注、商品行信创建采购收货单

输入参数：无

名称	类型	是否必须	描述
----	----	------	----

PurchaseOrderId, 采购订单 ID	Int32	No	PurchaseOrderId 与 PurchaseOrderNo 必须二选一
PurchaseOrderNo, 采购订单号	String	No	PurchaseOrderId 与 PurchaseOrderNo 必须二选一
WhsAreaCode, 库区代码	String	Yes	
Memo,备注	String	No	
UserDefinedField1 自定义字段 1	String 100	No	
UserDefinedField2 自定义字段 2	String 100	No	
UserDefinedField3 自定义字段 3	String 100	No	
UserDefinedField4 自定义字段 4	String 100	No	
UserDefinedField5 自定义字段 5	String 100	No	
UserDefinedField6 自定义字段 6	double	No	
UserDefinedField7 自定义字段 7	double	No	
UserDefinedField8 自定义字段 8	DateTime	No	
LineInfo, 收货行信息	List<PurchaseRecei ptLine>	Yes	PurchaseReceiptLine 对象包含 3 个属性： LineNum int 采购订单行号 Quantity int 商品数量 SerialNumbers string 序列号

返回结果：

如果收货成功，则返回空 Json。否则返回错误码和错误信息。

2.13 采购退货草稿查询

2.13.1 应用场景

根据开始修改时间和截至修改时间来查询系统中采购退货草稿数据。

2.13.2 规则

- 登录账户所需系统权限: 仓库管理/采购退货
- 查询必须提供采购退货单修改开始时间和截至时间参数。
- 必须采用倒序的分页方式（从最后一页往回取）才能避免漏单问题。

2.13.3 API 内容

基本信息：

API 名称	B1EC2.PurchaseReturn.Query
功能描述	获取商家所有采购退货草稿数据

输入参数：无

名称	类型	是否必须	描述
WhsId, 仓库 ID	Int32	No	默认查询所有仓库的数据，除了默认值外每次只能查询一个仓库
StartTime, 订单修改开始时间	DateTime	Yes	
EndTime, 订单修改结束时间	DateTime	Yes	
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int32	No	取值范围：1~100，默认值：50

返回结果：

名称	类型	描述
TotalResults, 搜索到的采购退货草稿总数	Int32	订单集合
PurchaseReturn 采购退货主表	Object	1) 出库单状态 PurchaseReturnStatus : 10: 草稿;20:已退货;30:取消
└─ ReturnId, 订单 ID(int)		
└─ PurchaseReturnNo, 系统订单号(string)		
└─ SupplierId, 供应商ID(int)		
└─ WhsId, 仓库ID(int)		
└─ PurchaseReturnStatus, 出库单状态(int)		

<ul style="list-style-type: none"> └ PostDate,退货时间(DateTime) └ ReopenPurchaseOrder , 是否打开原始订单(bool) └ OpenTotal , 未清金额(double) └ ReturnTotal , 退货金额(double) └ Memo , 备注(string , 254) └ CreateTime , 创建日期(DateTime) └ LastModifiedTime , 最后修改日期(DateTime) └ LastModifiedUser , 最后修改人(string , 20) └ CreateUser , 创建人(string , 20) └ PurchaseReturnLine 退货单行 <ul style="list-style-type: none"> └ ReturnId, 订单 ID(int) └ LineNum, 行号(int) └ BaseOrderId, 基础单据 ID(int) └ BaseLineId, 基本单据 LineId(int) └ BaseOrderNo, 基本单据 No(string) └ ItemId, 商品 ID(int) └ ItemName, 商品名称(string,100) Not Null └ SkuCode, 商品编码(string,50) 注 : 如果为 item 没有 sku,则 skucode 等于货号 └ SkuName, 规格名称(string,100) └ Price, 单价(double) └ Quantity, 数量(int) └ StockPrice, 库存成本(double) └ StockValue, 库存金额(double) └ LineTotal, 行总计(double) └ WhsAreaId, 库区 Id(int) └ LineMemo, 行备注(string,254) └ ItemInfoMemo, 商品备注(string , 254) └ BatchProdcutDate, 批次生产日期 (DateTime) └ BatchExpiryDate, 批次有效日期(DateTime) 		
---	--	--

└ BatchMemo, 批次备注(string , 100)		
└ NeedSerial, 是否序列号管理(bool)		
└ SerialNumbers, 序列号(string)		
└ Unit, 单位(string , 20)		
└ Proportion, 比例(int)		
└ Barcode, 商品条码(string , 254)		

JSON 范例 :

```
{
  "TotalResults": 1,
  "PurchaseReturns": [
    {
      "Supplier": {
        "SupplierId": 2,
        "SupplierCode": "V00001",
        "SupplierName": "测试供应商 1",
        "Enabled": true,
        "Province": "上海",
        "City": "上海市辖区",
        "District": "卢湾区",
        "Address": "钦州路 100 号",
      },
      "Warehouse": {
        "WhsCode": "10",
        "WhsName": "主仓库"
      },
      "Lines": [
        {
```

```
"ItemInfo": {  
    "ItemCode": "A10001",  
},  
"WhsArea": {  
    "WhsAreaId": 1,  
    "WhsAreaCode": "1010",  
    "WhsId": 1,  
    "AreaName": "次品区",  
},  
  
"ReturnId": 4,  
"LineNum": 0,  
"BaseOrderId": 10,  
"BaseLineId": 0,  
"BaseOrderNo": "30140704000010",  
"ItemId": 26,  
"SkuCode": "A1000111",  
"SkuName": "玫红色 ",  
"ItemName": "VOJO 正品彩色面条 iphone4/4s ipad1/2/3 数据线 苹果充电线  
短线",  
"Unit": "2121",  
"Quantity": 20,  
"StockPrice": 0,  
"StockValue": 0,  
"Price": 0,  
"LineTotal": 0,  
"WhsAreaId": 1,  
"LineMemo": "",  
"ItemInfoMemo": "",
```

```
        "BatchProductDate": null,  
        "BatchExpiryDate": null,  
        "BatchMemo": "",  
        "NeedSerial": false,  
        "SerialNumbers": "212 的萨达,w2eqe,rwqrewq",  
        "Proportion": 1  
    }  
],  
    "ReturnId": 4,  
    "PurchaseReturnNo": "34140704000004",  
    "SupplierId": 2,  
    "WhsId": 1,  
    "PurchaseReturnStatus": 10,  
    "PurchaseReturnStatusEnum": 10,  
    "PostDate": "2014-07-04T16:03:35",  
    "ReopenPurchaseOrder": false,  
    "CreateUser": "admin",  
    "CreateTime": "2014-07-04T16:03:35",  
    "LastModifiedTime": "2014-07-04T16:03:35",  
    "LastModifiedUser": "admin",  
    "Memo": "",  
    "OpenTotal": 0,  
    "ReturnTotal": 0  
} ]  
}
```

2.14 采购退货

2.14.1 应用场景

根据采购退货草稿 ID、库区代码、商品行信息进行采购退货

2.14.2 规则

- 登陆账户所需系统权限: 采购收货

2.14.3 API 内容

基本信息：

API 名称	B1EC2.Purchase.Return
功能描述	根据采购退货草稿 ID、库区代码、商品行信息进行采购退货

输入参数：无

名称	类型	是否必须	描述
ReturnId, 采购退货单 ID	Int32	No	ReturnId 与 PurchaseReturnNo 必须二选一
PurchaseReturnNo, 采购退货单号	String	No	ReturnId 与 PurchaseReturnNo 必须二选一
WhsAreaCode, 库区代码	String	Yes	
LineInfo, 收货行信息	List<PurchaseReturnLineInfo>	Yes	PurchaseReturnLineInfo 对象包含 3 个属性： LineNum int 采购订单行号 Quantity int 商品数量

			SerialNumbers string 序号号 WhsCode 库区代码
--	--	--	--

返回结果：

如果收货成功，则返回空 Json。否则返回错误码和错误信息。

2.15 宝贝查询

2.15.1 应用场景

查询电商大师中的定义的宝贝主数据。

2.15.2 规则

- 登陆账户所需系统权限: 宝贝管理
- 查询必须提供商品修改开始时间和截至时间参数。

2.15.3 API 内容

基本信息：

API 名称	B1EC2.ShopItem.Query
功能描述	获取商家所有宝贝主数据信息

输入参数：无

名称	类型	是否必须	描述
StartTime, 商品修改开始时间	String	Yes	格式为: 2012-05-12 16:58:00
EndTime, 商品修改结束时间	String	Yes	格式同上
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int33	No	取值范围：1~100，默认值：50
ShopId 店铺 id	String	No	

返回结果：

名称	类型	描述
TotalResults, 搜索到的宝贝总数	Int32	无
ShopItems 宝贝对象	Object	宝贝集合
└ ShopItemCode, 宝贝编码(String,50)		
└ ShopItemName, 宝贝名称(string,100)		
└ Type, 发布类型(string,5)		

<ul style="list-style-type: none"> └ ShopId, 店铺 Id (int) └ IsVirtual, 是否虚拟商品(bool) └ IsAgent, 是否代销(bool) └ IsViolative, 是否违规(bool) └ PictureURI, 主图 URL(string,254) └ OuterId, 商家编码(string,50) └ Quantity, 数量(int) └ Price, 商品价格(double) └ Size, 体积(double) └ Weight, 重量 (千克) (double) └ Status, 状态(string,20) “在售中” “在库中” └ Memo, 备注(string,254) └ CreatedTime, 创建日期(DateTime) └ UpdateTime, 最后修改日期(DateTime) └ UserSign, 修改人(string) └ Lines, ShopSku 行(list) 宝贝规格对象 <ul style="list-style-type: none"> └ ShopItemCode, 宝贝编码(string,50) └ ShopSkuCode,网店 SkuCode(string,20) └ OuterId, 商家代码(string,50) └ Property1, 销售属性 1(string,100) └ Property2, 销售属性 2(string,100) └ Properties, 淘宝属性(string,200) └ Price,价格(double) └ Size, 体积(立方米)(double) └ Weight, 重量 (千克) (double) └ Quantity, 商品数量(int) 		
--	--	--

返回结果：

```
{
  "TotalResults": 1,
  "ShopItems": [
    {
      "Shop": {
        "ShopName": "线下店",
        "ShopType": "Offline",
      },
      "Lines": [
        {
          "ShopItemCode": "10448767293",
          "ShopSkuCode": "30553370617",
          "OuterId": "4424-1",
          "Property1": "1500mm ( 35-348-2",
          "Property2": "50cm",
          "Properties": "1627207:28332;122216547:4023391",
          "Price": 176,
          "Size": 0,
          "Weight": 0,
          "Quantity": 100,
          "LinkError": null
        },
        {
          "ShopItemCode": "10448767293",
          "ShopSkuCode": "30553370618",
          "OuterId": "4352-1",
          "Property1": "600mm ( 35-346-23",
```

```
"Property2": "50cm",
"Properties": "1627207:3232481;122216547:4023391",
"Price": 26.9,
"Size": 0,
"Weight": 0,
"Quantity": 99,
"LinkError": null
},
{
"ShopItemCode": "10448767293",
"ShopSkuCode": "30553370619",
"OuterId": "4350-1",
"Property1": "150mm ( 35-344-23",
"Property2": "50cm",
"Properties": "1627207:3232483;122216547:4023391",
"Price": 5.6,
"Size": 0,
"Weight": 0,
"Quantity": 100,
"LinkError": null
},
{
"ShopItemCode": "10448767293",
"ShopSkuCode": "30553370620",
"OuterId": "0448-1",
"Property1": "300mm ( 35-345-23",
"Property2": "50cm",
"Properties": "1627207:3232484;122216547:4023391",
```

```
"Price": 13,
"Size": 0,
"Weight": 0,
"Quantity": 100,
"LinkError": null
},
{
  "ShopItemCode": "10448767293",
  "ShopSkuCode": "30553370621",
  "OuterId": "0465-1",
  "Property1": "1000mm ( 35-347-2",
  "Property2": "50cm",
  "Properties": "1627207:90554;122216547:4023391",
  "Price": 69,
  "Size": 0,
  "Weight": 0,
  "Quantity": 100,
  "LinkError": null
}
],
"ShopItemCode": "10448767293",
"ShopItemName": "史丹利 不锈钢直尺 规格可选 150mm-1500mm",
"Type": "一口价",
"ShopId": 7,
"IsVirtual": false,
"IsAgent": false,
"IsViolative": false,
"PictureUrl":
```

```
"http://img01.taobaocdn.com/bao/uploaded/i1/T1d186XcJdXXarhpE0_033840.jpg",  
    "OuterId": "BXGZC",  
    "Quantity": 499,  
    "Price": 5.6,  
    "Size": 0,  
    "Weight": 0,  
    "Status": "在售中",  
    "Memo": "",  
    "CreatedTime": "2011-05-21T21:24:22",  
    "UpdateTime": "2014-07-25T16:58:39",  
    "UserSign": "",  
    "LinkError": null  
    }  
]  
}
```

2.16 订单拆单

2.16.1 应用场景

根据订单 ID/订单 No、行号进行拆单。

2.16.2 规则

➤ 登陆账户所需系统权限: 销售管理/订单查询/

2.16.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.Split
功能描述	根据订单 ID/订单 No、行号进行拆单。

输入参数：无

名称	类型	是否必须	描述
OrderId, 订单 ID	Int32	No	OrderId 与 OrderNo 必须二选一
OrderNo, 订单号	String	No	OrderId 与 OrderNo 必须二选一
LineNums, 行号	String	Yes	多行行号以“,”（逗号）分割

返回结果：

如果发货成功，则返回空 Json。否则返回错误码和错误信息。

2.17 售后单更新

2.17.1 应用场景

售后单 ID，售后单号码更新售后单自定义字段、收货信息和备注。

2.17.2 规则

- 登陆账户所需系统权限: 售后创建/售后处理

2.17.3 API 内容

基本信息：

API 名称	B1EC2.AfterSalesService.Update
功能描述	更新售后单自定义字段、收货信息和备注

输入参数：

名称	类型	是否必须	描述
AfterSaleServiceId, 售后单 ID	Int32	No	AfterSaleServiceId 与 AfterSaleServiceNo 必须二选一
AfterSaleServiceNo, 售后单单号	String	No	AfterSaleServiceId 与 AfterSaleServiceNo 必须二选一
RefundMemo, 退回备注	String	No	
UserDefinedField1,自定义属性 1	String	No	
UserDefinedField2,自定义属性 2	String	No	
UserDefinedField3,自定义属性 3	String	No	

UserDefinedField4,自定义属性 3	String	No	
ReceiverName, 收货人姓名	String	No	
ReceiverState, 收货人省份	String	No	
ReceiverCity, 收货人城市	String	No	
ReceiverDistrict, 收货人地区	String	No	
ReceiverAddress, 收货人地址	String	No	
ReceiverZip, 收货人邮编	String	No	
ReceiverMobile, 收货人手机	String	No	
ReceiverPhone, 收货人电话	String	No	

返回结果：

如果成功，则返回空 Json。否则返回错误码和错误信息。

2.18 售后单创建

2.18.1 应用场景

根据销售订单创建售后单

2.18.2 规则

➤ 登陆账户所需系统权限: 售后创建

2.18.3 API 内容

基本信息：

API 名称	B1EC2.AfterSalesService.Create
功能描述	根据销售订单创建售后单

输入参数：

名称	类型	是否必须	描述
Type, 售后单类型	Int32	Yes	10:退货退款;20:仅退款; 30:换货; 40:疑难件; 50:补发货;60:维修
OrderId, 销售订单 Id	Int	No	OrderId 与 OrderNo 必须二选一
OrderNo, 销售订单号	String	No	OrderId 与 OrderNo 必须二选一
ReturnWhsId, 退货仓库 Id	Int	No	
DeliveryWhsId, 发货仓库 Id	Int	No	换货/补发货/维修 生成新订单的发货仓库
RefundReasonId, 售后原因 Id	Int	No	
RefundMemo, 退回备注	String	No	
RefundFeeTotal, 应退金额	Double	No	
FillupFee, 应补金额	Double	No	
ReturnExpressId, 退回快递公司 Id	Int	No	
ExpressTrackNo, 退回快递单号	String	No	
UserDefinedField1, 自定义属性 1	String	No	
UserDefinedField2, 自定义属性 2	String	No	
UserDefinedField3, 自定义属性 3	String	No	
UserDefinedField4, 自定义属性 4	String	No	
ReceiverName, 收货人姓名	String	No	不传默认按原订单的
ReceiverState, 收货人省份	String	No	不传默认按原订单的
ReceiverCity, 收货人城市	String	No	不传默认按原订单的
ReceiverDistrict, 收货人地区	String	No	不传默认按原订单的
ReceiverAddress, 收货人地址	String	No	不传默认按原订单的
ReceiverZip, 收货人邮编	String	No	不传默认按原订单的
ReceiverMobile, 收货人手机	String	No	不传默认按原订单的

ReceiverPhone, 收货人电话	String	No	不传默认按原订单的
ReturnLineInfo, 退货商品信息 数据格式: ReturnLineInfo = OrderLineNum1: Quantity1; OrderLineNum2: Quantity2	List<AfterSaleServiceReturnItemLine>	No	AfterSaleServiceReturnItemLine 对象包含 2 个属性: OrderLineNum int 订单行号 Quantity int 商品数量 备注: Type 为 10:退货退款;20:仅退款; 30: 换货; 40: 疑难件; 60: 维修时, 该参数为必须
ExchangeAndRedeliveryLineInfo, 换货/补发商品信息 数据格式: 同上	List<AfterSaleServiceExchangeAndRedeliveryItemLine>	No	AfterSaleServiceExchangeAndRedeliveryItemLine 对象包含 2 个属性: OrderLineNum int 订单行号 Quantity int 商品数量 备注: Type为 30: 换货; 50: 补发货时, 该参数为必须

返回结果: 如果创建成功, 则返回 AfterSaleServiceNo

2.19 售后原因查询

2.19.1 应用场景

查询商家目前在电商大师 ERP 中的所有售后原因。

2.19.2 规则

- 登陆账户所需系统权限: 售后创建

2.19.3 API 内容

基本信息：

API 名称	B1EC2.RefundReason.Query
功能描述	获取商家售后原因信息

输入参数：无

返回结果：

名称	类型	描述
RefundReasons 售后原因对象	Object	
└ ReasonId, 售后原因 ID(int)		
└ ReasonName, 原因名称(string,20)		
└ IsSystem, 是否系统原因(bool)		
└ LastModifiedTime, 最后修改时间(DateTime)		
└ LastModifiedUser, 最后修改人(String,20)		

JSON 范例：

```
{
  "RefundReasons": [
    {
      "ReasonId": 1,
      "ReasonName": "Reason1",
      "IsSystem": false,
      "LastModifiedTime": "2014-11-26T11:15:23",
      "LastModifiedUser": "admin"
    },
    {
      "ReasonId": 2,
      "ReasonName": " Reason2",
      "IsSystem": false,
      "LastModifiedTime": "2014-11-26T11:15:24",
      "LastModifiedUser": "admin"
    }
  ]
}
```

```

{
  "ReasonId": 3,
  "ReasonName": " Reason3",
  "IsSystem": false,
  "LastModifiedTime": "2014-11-26T11:15:24",
  "LastModifiedUser": "admin"
},
]
}

```

2.20 商品库存查询

2.20.1 应用场景

查询商家目前在电商大师 ERP 中的商品库存。

2.20.2 规则

- 登陆账户所需系统权限: 库存管理

2.20.3 API 内容

基本信息：

API 名称	B1EC2.Stock.Query
功能描述	获取商家商品库存

输入参数：无

WhsId, 仓库 Id	Int	No	不传默认返回所有仓库
SkuInfo, 商品信息	List<String >	Yes	

返回结果：

名称	类型	描述
Stocks 商品库存对象	Object	
└─ WhsId, 仓库 Id(int)		
└─ SkuCode, Sku 代码(string,50)		
└─ Quantity, 数量(double)		

JSON 范例：

```
{
  "Stocks": [
    {
      "WhsId": 1,
      "SkuCode": "test01001001",
      "Quantity": 10
    },
    {
      "WhsId": 1,
      "SkuCode": "test01001002",
      "Quantity": 1
    },
    {
      "WhsId": 1,
      "SkuCode": "test01001003",
      "Quantity": 19
    }
  ]
}
```

2.21 订单创建

2.21.1 应用场景

创建手工订单

2.21.2 规则

➤ 登陆账户所需系统权限: 手工开单

2.21.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.Create
功能描述	手工开单

输入参数：

名称	类型	是否必须	描述
ShopId, 店铺 Id	Int	Yes	
WhsId, 仓库 Id	Int	Yes	
ExpressId, 快递 Id	Int	Yes	
CustomerName, 客户名称	String 50	Yes	
ShopOrderNo 平台订单编号	String 254	No	
ActPostFee, 物流费用	Double	No	
SellerMemo, 订单备注	String 1000	No	
IsCod, 是否货到付款	Bool	No	默认 false
UserDefinedField1, 自定义属性 1	String 50	No	
UserDefinedField2, 自定义属性 2	String 50	No	
UserDefinedField3, 自定义属性 3	String 100	No	

UserDefinedField4, 自定义属性 4	String 100	No	
UserDefinedField5, 自定义属性 5	String 254	No	
UserDefinedField6, 自定义属性 6	Double	No	
UserDefinedField7, 自定义属性 7	Double	No	
UserDefinedField8, 自定义属性 8	DateTime	No	
UserDefinedField9, 自定义属性 9	String	No	
UserDefinedField10, 自定义属性 10	String	No	
UserDefinedField11, 自定义属性 11	String	No	
DiscountFee 折扣金额	Double	No	
OrderTotal 应付金额	Double	No	
ShopOrderStatus 平台订单状态	String	No	
ShopPayTime 网站付款时间	DateTime	No	
ReceiverName, 收货人姓名	String	No	
ReceiverState, 收货人省份	String	No	
ReceiverCity, 收货人城市	String	No	
ReceiverDistrict, 收货人地区	String	No	
ReceiverAddress, 收货人地址	String	No	
ReceiverZip, 收货人邮编	String	No	
ReceiverMobile, 收货人手机	String	No	
InvoiceType, 开票类型	Int	No	0: 无需发票; 10: 普通发票; 20: 增值税普通发票; 25: 增值税专用发票; 30: 收据; 默认0
InvoiceTotal, 开票金额	Double	No	
InvoiceExpressNo, 开票快递号	String	No	
InvoiceMemo, 开票备注	String	No	
InvoiceNo, 发票号码	String	No	
InvoiceName, 发票抬头	String	No	
CustomTax 税金	double	No	跨境单必填
CustomIdNo 身份证号	String 50	No	跨境单必填

CustomName 报关人	String 50	No	跨境单必填
ItemLineInfo,商品信息 数据格式 : ItemLineInfo = SkuCode1~Quantity1~Price1~LineMemo1~ LineUdf1~LineUdf2~isVirtua~parentSku^ SkuCode1~Quantity1~Price1~LineMemo1~ LineUdf1~ LineUdf2~isVirtual~parentSku^	List< SalesOrderC reateLine >	Yes	SalesOrderCreateLine 对象包含 6 个属性 : SkuCode string 规格编码 Quantity int 商品数量 Price double 价格 LineMemo String 行备注 LineUdf1 String 自定义字段 1 LineUdf2 double 自定义字段 2 isVirtual bool 是否虚拟 parentSku String 父商品

返回结果 : 如果创建成功, 返回 OrderNo

2.22 商品创建

2.22.1 应用场景

创建商品

2.22.2 规则

➤ 登陆账户所需系统权限: 商品资料

2.22.3 API 内容

基本信息：

API 名称	B1EC2.Item.Create
功能描述	创建商品

输入参数：

名称	类型	是否必须	描述
ItemCode, 货号	String 50	Yes	
ItemName, 商品名称	String 254	Yes	
CatCode, 类目代码	Int	No	必须是叶子节点类目
SupplierId, 供应商 Id	Int	No	
SupplierCode, 供应商货号	String 250	No	
PictureUrl 主图 Url	String 254	No	
BarCode, 条码	String 100	No	
PurchasePrice, 采购价格	Double	No	
SalesPrice, 销售价格	Double	No	
LowestPrice, 最低售价	Double	No	
MarketPrice, 吊牌价格	Double	No	
Property1, 自定义属性 1	String 100	No	
Property2, 自定义属性 2	String 100	No	
Property3, 自定义属性 3	String 100	No	
Property4, 自定义属性 4	String 100	No	
Property5, 自定义属性 5	String 100	No	
Property6, 自定义属性 6	String 100	No	
Property7, 自定义属性 7	String 100	No	
Property8, 自定义属性 8	String 100	No	

Property9, 自定义属性 9	String 100	No	
Property10, 自定义属性 1 0	String 100	No	
Property11, 自定义属性 1 1	String 100	No	
Property12, 自定义属性 1 2	String 100	No	
Unit, 单位	String 20	No	
PurchaseUnit, 采购单位	String 20	No	
Size, 体积 (立方米)	double	No	
Weight, 重量 (千克)	double	No	
IsVirtual, 是否虚拟商品	Bool	No	默认 false
SkuInfo, Sku 信息 数据格式 : SkuInfo = SkuCode:BarCode: Property1: Property2: SalesPrice: Unit^SkuCode:BarCode: Property1: Property2: SalesPrice: Uni	List< SkuCreateLi ne >	No	SkuCreateLine 对象包含 6 个属性 : SkuCode string 50 SkuCode BarCode int 条形码 Property1 string 100 销售属 性 1 Property2 string 100 销售属性 2 SalesPrice double 售价 Unit string 20 单位

返回结果 : 如果创建成功, 返回 ItemId

2.23 组合商品创建

2.23.1 应用场景

创建组合商品

2.23.2 规则

- 登陆账户所需系统权限: 组合商品

2.23.3 API 内容

基本信息：

API 名称	B1EC2.Bom.Create
功能描述	创建组合商品

输入参数：

名称	类型	是否必须	描述
SkuCode, 规格编码	String 50	Yes	
BomLineInfo, Bom 行信息 数据格式：BomLineInfo=SkuCode:Quantity; SkuCode:Quantity;SkuCode:Quantity;	List< BomCreateLine >	Yes	BomCreateLine 对象包含 2 个属性： SkuCode string 50 SkuCode Quantity int 数量

返回结果：如果创建成功，返回 BomId

2.24 疑难件创建

2.24.1 应用场景

疑难件入库

2.24.2 规则

➤ 登陆账户所需系统权限: 售后创建

2.24.3 API 内容

基本信息：

API 名称	B1EC2.UnknownPackage.Create
功能描述	疑难件入库

输入参数：

名称	类型	是否必须	描述
ReturnWhsId, 退货仓库 Id	Int	Yes	
ReturnExpressId, 退回快递公司 Id	Int	No	
ExpressTrackNo, 退回快递单号	String	No	
UserDefinedField1, 自定义属性 1	String	No	
UserDefinedField2, 自定义属性 2	String	No	
UserDefinedField3, 自定义属性 3	String	No	
UserDefinedField4, 自定义属性 4	String	No	
ItemLineInfo, 入库商品明细 数据格式：ItemLineInfo= SkuCode:Quantity;	List<UnknownPackageItemLine	Yes	UnknownPackageItemLine 对象包含 2 个属性：

SkuCode:Quantity;	>		SkuCode string 规格编码 Quantity int 商品数量
-------------------	---	--	--

返回结果：如果创建成功，则返回 AfterSaleServiceNo

2.25 订单更新

2.25.1 应用场景

根据订单 ID 更新订单备注,拦截原因,自定义字段等

2.25.2 规则

➤ 登陆账户所需系统权限: 销售管理

2.25.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.Update
功能描述	根据订单 ID 更新订单备注,拦截原因,自定义字段等

输入参数：

名称	类型	是否必须	描述
OrderId, 订单 ID	Int32	No	OrderId 与 OrderNo 必须二选一
OrderNo, 订单号	String	No	OrderId 与 OrderNo 必须二选一
OrderMemo	String	No	
ReasonId ,拦截原因 Id	Int32	No	ReasonId 设置为 0,则取消拦截
HoldingMemo, 拦截备注	String	No	

UserDefinedField1	String	No	
UserDefinedField2	String	No	
UserDefinedField3	String	No	
UserDefinedField4	String	No	
UserDefinedField5	String	No	
UserDefinedField6	double	No	
UserDefinedField7	double	No	
UserDefinedField8	DateTime	No	
UserDefinedField9	String	No	
UserDefinedField10	String	No	
UserDefinedField11	String	No	

返回结果：

如果发货成功，则返回空 Json。否则返回错误码和错误信息。

2.26 账单查询

2.26.1 应用场景

根据对账开始时间和对账截至时间来查询系统中账单数据。

2.26.2 规则

- 登陆账户所需系统权限: 账单管理
- 查询必须提供开始时间和截至时间参数。

2.26.3 API 内容

基本信息：

API 名称	B1EC2.FinJournal.Query
功能描述	获取商家所有账单数据

输入参数：

名称	类型	是否必须	描述
ShopId, 店铺 Id	Int32	No	默认查询所有店铺的数据
StartTime, 账单对账开始时间	String	Yes	格式为: 2012-05-12 16:58:00
EndTime, 账单对账结束时间	String	Yes	格式同上
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int32	No	取值范围: 1~100, 默认值: 50

返回结果：

TotalResults 结果总数 int

FinJournal 账单表

- └─ FinAccount, 支付宝科目
 - └─ AccountCode, 科目编码(String)
 - └─ Type, 科目类型(int)
 - └─ AccountName,科目名称(string)
 - └─ Memo,备注(string)
- └─ BusinessId 业务流水号 string
- └─ ShopOrderId 订单号 string
- └─ ShopId 店铺 Id int
- └─ AccountCode 科目编码 string
- └─ Amount 交易金额 double
- └─ AmountBalance 账号余额 double
- └─ TransferStatus 账单状态 int
- └─ PostDate 应过账日期 DateTime
- └─ CreateTime 生成时间 DateTime
- └─ Memo 备注 string
- └─ LastModifiedTime 最后修改时间 DateTime
- └─ LastModifiedUser 最后修改人 string

JSON 范例：

```
{
  "TotalResults": 1,
  "FinJournals": [
    {
      "Shop": {
        "ShopId": 1,
        "ShopName": "Willey 淘宝",
```

```
    },  
    "FinAccount": {  
        "AccountCode": "100101",  
        "AccountName": "交易收款 01",  
    },  
    "FinJournalId": 1,  
    "BusinessId": "testBusinessId",  
    "ShopOrderId": "TestOrderId",  
    "ShopId": 1,  
    "AccountCode": "100101",  
    "Amount": 11.212,  
    "AccountBalance": 1,  
    "TransferStatus": 10,  
    "PostDate": "2015-02-02T14:49:54",  
    "CreateTime": "2015-02-02T14:49:57",  
    "Memo": "testMemo",  
    "LastModifiedTime": "2015-02-04T14:50:02",  
    "LastModifiedUser": "admin"  
    }  
    ]  
}
```

2.27 商品类目查询

2.27.1 应用场景

查询商家目前在电商大师 ERP 中的所有商品类目信息。

2.27.2 规则

- 登陆账户所需系统权限: 商品资料

2.27.3 API 内容

基本信息：

API 名称	B1EC2.Category.Query
功能描述	获取商家商品类目信息

输入参数：无

返回结果：

名称	类型	描述
Category 类目对象 └─ CatCode, 类目代码(int) └─ CatName, 类目名称(string,50) └─ FatherCode, 父类代码(int) └─ CatPath, 类目路径(string,100) └─ IsLeaf, 是否为叶子(bool)	Object	

JSON 范例：

```
{
  "Categories": [
    {
      "CatCode": 1,
      "CatName": "服饰",
      "FatherCode": null,
      "CatPath": "\\1",
      "IsLeaf": false
    },
    {
```

```
"CatCode": 2,  
  "CatName": "衣服",  
  "FatherCode": 1,  
  "CatPath": "\\1\\2",  
  "IsLeaf": true  
},  
{  
  "CatCode": 3,  
  "CatName": "裤子",  
  "FatherCode": 1,  
  "CatPath": "\\1\\3",  
  "IsLeaf": true  
}  
]  
}
```

2.28 组合商品查询

2.28.1 应用场景

查询电商大师中的定义的组合商品主数据。

2.28.2 规则

- 登陆账户所需系统权限: 商品管理/组合商品
- 查询必须提供商品修改开始时间和截至时间参数。

2.28.3 API 内容

基本信息：

API 名称	B1EC2.Bom.Query
功能描述	获取商家所有组合商品主数据信息

输入参数：无

名称	类型	是否必须	描述
StartTime,修改开始时间	String	Yes	格式为: 2012-05-12 16:58:00
EndTime,修改结束时间	String	Yes	格式同上
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int33	No	取值范围: 1~100, 默认值: 50

返回结果：

名称	类型	描述
----	----	----

└ ItemName, 商品名称

<p>TotalResults, 搜索到的组合商品总数</p> <p>Boms 组合商品对象</p> <ul style="list-style-type: none"> └ BomId, 组合商品 ID(int) └ ItemId, 商品 ID(int) └ ItemInfo, 商品 <ul style="list-style-type: none"> └ ItemCode, 商品代码 string └ ItemName, 商品名称 string └ ItemCode, 商品代码 string └ ItemName, 商品名称 string └ SkuCode, SkuCode string └ SkuName, 规格名称 string └ Unit, 单位 string └ StockNum, 库存数量 int └ SalesPrice, 销售价格 double └ Unit, 单位 string └ Memo, 备注 string └ LastModifiedTime, 最后修改时间 DateTime └ LastModifiedUser, 最后修改人 string └ Lines,Bomline (list) Bomline 对象 <ul style="list-style-type: none"> └ BomId, BomId ID(int) └ LineNum, 行号 int └ LineItemId, 商品 Id int └ SkuCode, SkuCode string └ ItemName, 商品名称 string └ SkuName, 规格 string └ Quantity, 数量 int └ Unit, 单位 string 	<p>Int32</p> <p>Object</p>	<p>无</p> <p>组合商品集合</p>
--	----------------------------	------------------------

└ SalesPrice , 销售价格 double		
----------------------------	--	--

```
{
  "TotalResults": 1,
  "Boms": [
    {
      "Lines": [
        {
          "BomId": 1,
          "LineNum": 0,
          "LineItemId": 2,
          "SkuCode": "test02001001",
          "ItemName": "test02Name",
          "SkuName": "001 001",
          "Quantity": 5,
          "Unit": "",
          "SalesPrice": 0,
          "isVritual": false
        },
        {
          "BomId": 1,
          "LineNum": 1,
          "LineItemId": 2,
          "SkuCode": "test02001002",
          "ItemName": "test02Name",
          "SkuName": "001 002",
          "Quantity": 10,
          "Unit": "",
          "SalesPrice": 0,
          "isVritual": false
        }
      ],
      "BomId": 1,
      "ItemId": 1,
      "ItemCode": "test01",
      "ItemName": "test01Name",
      "SkuCode": "test01001001",
      "SkuName": "001 001",
      "Unit": "",
      "StockNum": 0,
      "SalesPrice": 0,
      "Memo": "",
      "LastModifiedTime": "2015-03-11T17:34:45",
    }
  ]
}
```

```

    "LastModifiedUser": "admin"
  }
]
}

```

2.29 平台订单推送

2.29.1 应用场景

创建平台订单

2.29.2 规则

➤ 登陆账户所需系统权限: 手工开单

2.29.3 API 内容

基本信息：

API 名称	B1EC2.ShopOrder.Push
功能描述	平台订单创建

输入参数：

名称	类型	是否必须	描述
ShopOrderNo, 平台订单号	String 100	Yes	
ShopId, 店铺 Id	Int	Yes	
MemberNick, 客户名称	String 25	Yes	
OrderStatus, 订单状态	Int	Yes	0:草稿 10:未发货 20:已发货 30:已完结 40:已关闭 50:已取消
ShopCreatedTime, 下单时间	DateTime	Yes	
OrderStatusName, 平台订单状态	String 10	No	平台订单状态描述, 如已付款, 等待发货等等。都为中文描述, 仅备注作用
IsCod, 是否货到付款	Bool	No	默认 false
IsDistribution, 是否分销	Bool	No	默认 false

IsJhs,是否聚划算	Bool	No	默认 false
IsPresale, 是否预售	Bool	No	默认 false
IsMobile, 是否手机订单	Bool	No	默认 false
DiscountFee, 折扣金额	Double	No	
PostFee, 邮费	Double	No	
AdjustFee, 调整金额	Double	No	
GoodsTotal, 商品总额	Double	No	
OrderTotal, 应付金额	Double	No	
ReceivedTotal, 实际收款	Double	No	
ShopPayTime, 平台付款时间	DateTime	No	
BuyerMemo, 买家留言	String 1000	No	
SellerMemo, 卖家备注	String 1000	No	
ShopFlag, 卖家旗帜	String 6	No	合法值为: blue, green, orange, pink, purple, red, yellow
ReceiverName, 收货人姓名	String 25	No	
ReceiverState, 收货人省份	String 25	No	
ReceiverCity, 收货人城市	String 25	No	
ReceiverDistrict, 收货人地区	String 25	No	
ReceiverAddress, 收货人地址	String 100	No	
ReceiverZip, 收货人邮编	String 25	No	
ReceiverMobile, 收货人手机	String 25	No	
ReceiverPhone, 收货人电话	String 25	No	
InvoiceType, 开票类型	Int	No	0:无需发票;10:普通发票;20:增值税普通发票;25:增值税专用发票;30:收据; 默认0
InvoiceMemo,开票备注	String 254	No	
InvoiceName,发票抬头	String 100	No	

ExpressName, 快递公司	String 15	No	
ExpressTrackNo, 快递单号	String 20	No	
BuyerAlipayNo, 买家支付宝	String 100	No	
BuyerEmail, 买家邮箱	String 50	No	
AlipayOrderNo, 支付宝付款编号	String 35	No	
DeliveryTime, 发货时间	DateTime	No	
EndTime, 交易完成时间	DateTime	No	
UserDefinedField1, 自定义字段 1	String 50	No	
UserDefinedField2, 自定义字段 2	String 254	No	
CustomTax 税金	double	No	
CustomIdNo 身份证号	String 50	No	跨境单必填
CustomName 报关人	String 50	No	跨境单必填
CustomPaymentName 海关支付方式	String 50	No	跨境单必填。与付款信息中的支付方式 ID 对应的支付方式相同。
CustomPaymentNo 海关支付单号	String 20	No	跨境单必填。与付款信息中的支付单号相同。
ItemLineInfo,商品信息 数据格式 ItemLineInfo= ShopLineNo: OuterId:Quantity:Price:LineUdf1:LineUdf2: ItemName:Skuname: LineTotal:LineCustomTax:LineCustomTotal; DiscountLineInfo 和 PaymentLineInfo 一样	List< ShopOrderCreateLine >	Yes	ShopOrderCreateLine 对象包含 11 个属性： ShopLineNo string 50 平台订单行号 OuterId string 100 外部代码 Quantity int 数量 Price double 价格 (含税价格) LineUdf1 String 50 自定义字段 1 LineUdf2 String 50 自定义字段 2 ItemName String 商品名称 Skuname String 规格名称 LineTotal 含税行总价 如果没有填 0

			LineCustomTax 行税金 如果没有填 0 LineCustomTotal 不含税总价 如果没有填 0
DiscountLineInfo,折扣信息	List<ShopOrderCreateDiscount>	No	ShopOrderCreateDiscount 对象包含 2 个属性： DiscountName string 30 折扣名称 DiscountFee double 折扣额
PaymentLineInfo,付款信息	List<ShopOrderCreatePayment>	No	ShopOrderCreatePayment 对象包含 2 个属性： PaymentId int 付款方式 (1:现金支付 2:银行转账 3:网上支付 4:支付宝 5:天猫积分 6:货到付款 7:账户支付 26:微信支付 可以参考附录) PaymentTotal double 付款金额 PaymentNo string 付款单号

返回结果：如果创建成功，返回 ShopOrderNo

2.30 客户创建

2.30.1 应用场景

创建客户

2.30.2 规则

➤ 登陆账户所需系统权限: 会员管理

2.30.3 API 内容

基本信息：

API 名称	B1EC2.Customer.Create
功能描述	客户创建

输入参数：

名称	类型	是否必须	描述
CustomerName, 客户名称	String 50	Yes	
ShopId, 店铺 Id	Int	Yes	
Mobile, 手机	String 30	No	
ReceiveName, 收货人姓名	String 30	No	
Province, 省份	String 32	No	
City, 市	String 32	No	
District, 区(县)	String 32	No	默认 false
Address, 详细地址	String 228	No	默认 false
Email, 邮箱	String 100	No	
ZipCode, 邮编	String 6	No	
Memo, 备注	String 254	No	
BuyerCredit, 买家信誉	String20	No	
LevelId, 会员等级	Int	No	
IsFenXiao, 是否分销商	Bool	No	
UserDefinedField1, 自定义字段 1	String 50	No	
UserDefinedField2, 自定义字段 2	String 50	No	
UserDefinedField3, 自定义字段 3	String 50	No	
UserDefinedField4, 自定义字段 4	String 50	No	
UserDefinedField5, 自定义字段 5	String 50	No	

UserDefinedField6, 自定义字段 6	String 50	No	
CustomerCode 客户编码	String	No	如果没有填写客户编码, 系统会自动生成

返回结果: 如果创建成功, 返回 CustomerCode

2.31 宝贝推送

2.31.1 应用场景

宝贝推送

2.31.2 规则

➤ 登陆账户所需系统权限: 宝贝管理

2.31.3 API 内容

基本信息:

API 名称	B1EC2.ShopItem.Push
功能描述	宝贝推送

输入参数:

名称	类型	是否必须	描述
ShopItemCode, 宝贝编码	String 50	Yes	
ShopId, 店铺 Id	Int	Yes	

Status,上架状态	String 20	Yes	合法值： 在售中 ；在库中
CreatedTime, 创建时间	DateTime	Yes	
UpdateTime, 修改时间	DateTime	Yes	
ShopItemName, 宝贝名称	String 100	No	
IsVirtual, 是否虚拟商品	Bool	No	默认false
PictureUrl, 主图 Url	String 254	No	
OuterId, 商家编码	String 50	No	
Quantity, 数量	Int	No	
Price, 价格	Double	No	
Size, 体积 (立方米)	Double	No	
Weight, 重量 (千克)	Double	No	
Memo, 备注	String 254	No	
SkusInfo, sku 信息	List< ShopSkuPush Line >	No	ShopSkuPushLine 对象包含 8 个属性： ShopSkucode string 50 网店 skucode OuterId String 50 商家代码 Property1 String 100 销售属 性 1 Property2 String 100 销售属 性 2 Price double 价格 Size double 体积 (立方米) Weight double 重量 (千克) Quantity int 数量

返回结果：如果创建成功，返回 ShopItemCode

2.32 转储单查询

2.32.1 应用场景

转储单查询

2.32.2 规则

- 登陆账户所需系统权限: 库存调拨

2.32.3 API 内容

基本信息：

API 名称	B1EC2.StockTransfer.Query
功能描述	转储单查询

输入参数：

名称	类型	是否必须	描述
StartTime, 转储单创建开始时间	String	Yes	格式为: 2012-05-12 16:58:00
EndTime, 转储单创建结束时间	String	Yes	格式同上
PageNum, 页码	Int32	No	取值范围:大于零的整数;默认值:1。
PageSize, 每页条数	Int33	No	取值范围 : 1~100 , 默认值 : 50
StockTransferNo, 转储单号	String	Yes	
StockTransferType, 转储类型	Int32	No	10 : 仓内调拨 20: 仓外调拨
StockTransferStatus, 转储状态	Int32	No	

返回结果：

名称	类型	描述
TotalResults, 搜索到的转储单数	Int32	无

StockTransfers, 转储单对象	Object	转储单对象
<ul style="list-style-type: none"> └─ Lines,转储单行(list) 转储单行对象 <ul style="list-style-type: none"> └─ TransferId, 转储单 ID(int) └─ LineNum, 行号(int) └─ ItemId, 商品 ID(int) └─ ItemInfo, 商品信息对象(list) └─ SkuCode, 商品编码(string) └─ SkuName, 商品规格(string) └─ ItemName, 商品名称(string) └─ Unit, 单位(string) └─ Quantity, 数量(int) └─ FromWhsArea, 从库区 ID(int) └─ WhsAreaFrom, 库区对象(list) └─ FromWhsAreaName, 库区名称(string) └─ ToWhsArea, 至库区 ID(int) └─ WhsAreaTo, 库区对象(list) └─ ToWhsAreaName, 至库区名称(string) └─ LineMemo, 备注(string) └─ SerialNumbers, 序列号(string) └─ TransferId, 转储单 ID(int) └─ StockTransferNo, 转储单号(string) └─ StockTransferType, 转储类型(int) └─ StockTransferTypeEnum, 转储类型枚举(enum) └─ StockTransferStatus, 转储状态(int) └─ StockTransferStatusEnum, 转储状态枚举(enum) └─ FromWhsId, 从仓库 ID(int) └─ WarehouseFrom, 仓库对象(list) └─ FromWhsAreaId, 从库区 ID(int) 		

└ ToWhsId, 至仓库 ID(int)		
└ WarehouseTo, 仓库对象(list)		
└ ToWhsAreaId, 至库区 ID(int)		
└ PostDate, 转储时间(datetime)		
└ CreateUser, 创建人(string)		
└ CreateTime, 创建时间(datetime)		
└ Memo, 备注(string)		

JSON 范例 :

```
{
  "TotalResults": 1,
  "StockTransfers": [
    {
      "Lines": [
        {
          "TransferId": 17,
          "LineNum": 0,
          "ItemId": 10,
          "ItemInfo": null,
          "SkuCode": "NB0533xxl01",
          "SkuName": "深紫色 XXL",
          "ItemName": "测试衬衫 3",
          "Unit": null,
          "Quantity": 2,
          "FromWhsArea": 3,
          "WhsAreaFrom": null,
          "FromWhsAreaName": null,
          "ToWhsArea": 4,
          "WhsAreaTo": null,
          "ToWhsAreaName": null,
          "LineMemo": "222",
          "SerialNumbers": null
        },
        {
          "TransferId": 17,
          "LineNum": 1,
          "ItemId": 5,
```

```
        "ItemInfo": null,  
        "SkuCode": "SP10001001",  
        "SkuName": "军绿色 170/88(M)",  
        "ItemName": "立领优衣库羽绒服男外套茄克",  
        "Unit": null,  
        "Quantity": 12,  
        "FromWhsArea": 3,  
        "WhsAreaFrom": null,  
        "FromWhsAreaName": null,  
        "ToWhsArea": 4,  
        "WhsAreaTo": null,  
        "ToWhsAreaName": null,  
        "LineMemo": "ggg",  
        "SerialNumbers": null  
    }  
],  
    "TransferId": 17,  
    "StockTransferNo": "34150423000017",  
    "StockTransferType": 10,  
    "StockTransferTypeEnum": 10,  
    "StockTransferStatus": 30,  
    "StockTransferStatusEnum": 30,  
    "FromWhsId": 1,  
    "WarehouseFrom": null,  
    "FromWhsAreaId": 0,  
    "ToWhsId": 1,  
    "WarehouseTo": null,  
    "ToWhsAreaId": 0,  
    "PostDate": "2015-03-20T00:00:00",  
    "CreateUser": "admin",  
    "CreateTime": "2015-04-23T11:17:13",  
    "Memo": ""  
} ] }
```

2.33 转储单行更新

2.33.1 应用场景

转储单查询

2.33.2 规则

- 登陆账户所需系统权限: 库存调拨修改

2.33.3 API 内容

基本信息：

API 名称	B1EC2.StockTransferLine.Update
功能描述	转储单行更新

输入参数：

名称	类型	是否必须	描述
TransferId, 转储单 ID	Int32	No	TransferId 和 StockTransferNo 必须二选一
StockTransferNo, 转储单号	String	No	TransferId 和 StockTransferNo 必须二选一
Memo,备注	String	No	
TransferLines, 库存转储单行信息	List<StockTransferLine>	No	StockTransferLine 对象包含 5 个属性: SkuCode : 商品编码 FromWhsArea : 从库区 ID ToWhsArea : 至库区 ID Quantity : 数量 LineMemo: 备注

返回结果：

如果成功，则返回空 Json。否则返回错误码和错误信息。

2.34 订单取消

2.34.1 应用场景

订单关闭

2.34.2 规则

➤ 登陆账户所需系统权限: 销售管理

2.34.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.Cancel
功能描述	根据订单 ID 或者订单号取消订单

输入参数：

名称	类型	是否必须	描述
OrderId, 订单 ID	Int32	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一
OrderNo, 订单号	String	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一
ShopOrderNo 平台订单号	String	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一

返回结果：

如果取消成功，则返回空 Json。否则返回错误码和错误信息。

注意：被取消订单必须未发货状态。

2.35 支付方式查询

2.35.1 应用场景

支付方式查询

2.35.2 规则

- 登陆账户所需系统权限: 销售管理

2.35.3 API 内容

基本信息：

API 名称	B1EC2.PaymentMethod.Query
功能描述	查询所有的支付方式

输入参数：无

返回结果：

名称	类型	描述
PaymentMethods 支付方式对象	Object	支付方式集合
└─ PayMentId, 支付方式 ID(int)		
└─ PayMentName, 支付方式名称(string,50)		

JSON 范例:

```
{
  "PaymentMethods": [
    {
      "PaymentId": 1,
      "PaymentName": "现金支付"
    }
  ]
}
```

```
},  
{  
  "PaymentId": 2,  
  "PaymentName": "银行转账"  
},  
{  
  "PaymentId": 3,  
  "PaymentName": "网上支付"  
},  
{  
  "PaymentId": 4,  
  "PaymentName": "支付宝"  
},  
{  
  "PaymentId": 5,  
  "PaymentName": "天猫积分"  
},  
{  
  "PaymentId": 6,  
  "PaymentName": "货到付款"  
},  
{  
  "PaymentId": 7,  
  "PaymentName": "账户支付"  
},  
{  
  "PaymentId": 8,  
  "PaymentName": "抵用卷"
```

```
    },  
    {  
        "PaymentId": 9,  
        "PaymentName": "其他方式"  
    },  
    {  
        "PaymentId": 10,  
        "PaymentName": "集分宝"  
    }  
]  
}
```

2.36 杂项出入库类型查询

2.36.1 应用场景

杂项出入库类型查询

2.36.2 规则

- 登陆账户所需系统权限: 库存管理

2.36.3 API 内容

基本信息：

API 名称	B1EC2.Goods.PassInAndOutType.Query
功能描述	查询所杂项出入库类型

输入参数：

名称	类型	是否必须	描述
Type, 出库类型	Int50	Yes	10 杂项入库 20 杂项出库

返回结果：

名称	类型	描述
GoodsPassInAndOutType 杂项出入库类型 └─ typeId, 出入库 ID(int) └─ typeName, 出入库名称(string,50)	Object	出入库集合

JSON 范例:

```

{
  "GoodsPassInAndOutType": [
    {
      "typeId": 1,
      "typeName": "杂项入库"
    },
    {
      "typeId": 2,
      "typeName": "采购入库"
    }
  ]
}
  
```


2.37 杂项出入库创建

2.37.1 应用场景

杂项出入库操作

2.37.2 规则

➤ 登陆账户所需系统权限: 仓库管理

2.37.3 API 内容

基本信息：

API 名称	B1EC2.Goods.PassInAndOut.Create
功能描述	杂项出入库创建

输入参数：

名称	类型	是否必须	描述
PassInAndOutId, 出入库类型 Id	Int	Yes	Id 可以通过 2.36 接口查询
PassInAndOutType, 出入库类型	Int	Yes	10 出库 20 入库
WhsId, 仓库 Id	Int	Yes	
WhsAreaCode, 库区编码	String 20	Yes	
PassInAndOutTime, 出入库时间	String 100	No	格式为: 2012-05-12 16:58:00,默认为当前时间
UserDefinedField1 自定义字段 1	String 100	No	入库时可填写, 也可不填写, 出库时不需要
UserDefinedField2 自定义字段 2	String 100	No	入库时可填写, 也可不填写, 出库时不需要
UserDefinedField3 自定义字段 3	String 100	No	入库时可填写, 也可不填写, 出库时不需要
UserDefinedField4 自定义字段 4	String 100	No	入库时可填写, 也可不填写, 出库时不需要
LineInfo, 出入库行明细	List< PassInAndOutLine	Yes	PassInAndOutLine 对象包含 5 个属性： SkuCode string 50 SkuCode

	>		Quantity int 数量 Unit string 100 单位 Price double 100 价格 Mark string 100 备注
--	---	--	--

返回结果：如果创建成功，返回{}

2.38 订单配货

2.38.1 应用场景

将订单状态修改为已配货，如果有跨境包，系统会自动推送海关。

2.38.2 规则

➤ 登陆账户所需系统权限: 销售管理

2.38.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.PickOut
功能描述	根据订单 ID 或者订单号执行配货

输入参数：

名称	类型	是否必须	描述
OrderId, 订单 ID	Int32	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一
OrderNo, 订单号	String	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一
ShopOrderNo 平台订单号	String	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一

返回结果：

如果配货成功，则返回空 Json。否则返回错误码和错误信息。

注意：被配货订单必须为待审核或者已审核状态。

2.39 订单打回待审

2.39.1 应用场景

将订单状态修改为未审核状态。

2.39.2 规则

➤ 登陆账户所需系统权限: 销售管理

2.39.3 API 内容

基本信息：

API 名称	B1EC2.SalesOrder.Unapprove
功能描述	根据订单 ID 或者订单号执行打回待审

输入参数：

名称	类型	是否必须	描述
OrderId, 订单 ID	Int32	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一
OrderNo, 订单号	String	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一
ShopOrderNo 平台订单号	String	No	OrderId 与 OrderNo 与 ShopOrderNo 必须三选一

返回结果：

如果配货成功，则返回空 Json。否则返回错误码和错误信息。

注意：订单必须未发货方可打回待审。

2.40 采购订单创建

2.40.1 应用场景

创建采购订单

2.40.2 规则

➤ 登陆账户所需系统权限: 采购管理/采购订单

2.40.3 API 内容

基本信息：

API 名称	B1EC2.PurchaseOrder.Create
功能描述	创建采购订单

输入参数：

名称	类型	是否必须	描述
SupplierId, 供应商 ID	Int32	Yes	
WhsId, 仓库 ID	String	Yes	
OrderStatus, 平台订单状态	String	Yes	10:未到货;20:已完成;30:已取消;40:草稿;
Memo, 备注	String	No	
UserDefinedField1 自定义字段1	String	No	
UserDefinedField2 自定义字段 2	String	No	
UserDefinedField3 自定义字段 3	String	No	
UserDefinedField4 自定义字段 4	String	No	

UserDefinedField5 自定义字段 5	String	No	
UserDefinedField6 自定义字段 6	Double	No	
UserDefinedField7 自定义字段 7	Double	No	
UserDefinedField8 自定义字段 8	DateTime	No	
PurchaseOrderLine 订单行信息 格式： LineStatus: SkuCode: SkuName: Price: Quantity: LineTotal: LineMemo: Unit: Barcode: UserDefinedField1;	List	Yes	共 10 个属性 └ LineStatus, 行状态(int) 注 : 10:未清;20:已清; └ SkuCode, 商品编码(string,50) └ SkuName, 规格名称(string,100) └ Price, 单价(double) └ Quantity, 数量(int) └ LineTotal, 行总计(string) └ LineMemo, 行备注(string,254) └ Unit, 单位(string , 20) └ Barcode, 商品条码(string , 254) └ UserDefinedField1 自定义字段 1 (string,50)

返回结果：

如果配货成功，则返回订单号 OrderNo。否则返回错误码和错误信息。

注意：订单必须未发货方可打回待审。

附录：对应关系

支付方式

ID	name
1	现金支付
2	银行转账
3	网上支付
4	支付宝
5	天猫积分
6	货到付款
7	账户支付
8	抵用卷
9	其他方式
10	集分宝
11	系统预留 2
12	系统预留 3
13	系统预留 4
14	系统预留 5
15	银联在线
16	盛付通
17	建设银行
18	中国银行
19	易付宝
20	农业银行
21	京东网银在线
22	国际支付宝
23	甬易支付
24	富友支付
25	连连支付
26	微信支付
27	快钱
28	网易宝
29	银盈通支付
30	鄞州银行
31	智惠支付
32	拉卡拉
33	北京银联
34	杭州银行
35	银联网络
36	重庆易极付
37	易宝支付
38	广州银联
39	上海银联

40	通联支付
41	民生银行
42	浙江银商
43	平安银行
44	首信易支付
45	百度钱包
46	易票联支付
47	浙江农信
48	招商银行