**数据接口以及json示例**

|  |  |
| --- | --- |
| 数据表 | |
| 1 | 用户表 |
| 2 | 用户地址 |
| 3 | 产品表 |
| 4 | 产品类型表 |
| 5 | 商品表 |
| 6 | 订单表 |
| 7 | 订单详情表 |
| 8 | 我的收藏 |
| 9 | 配送人员信息表 |
| 10 | 配送信息表 |
| 11 | 活动表 |
| 12 | 产品评价表 |
| 13 | 订单评价表 |
| 14 | 派送表 |
| 15 | 美食教程表 |
| 16 | 用户协议表 |
| 17 | 用户使用指南表 |
| 18 | 购物车表 |

# 用户API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| userId | |  |  | | --- | --- | | Number |  | | NOT NULL | 10001 | 用户数字ID PK |
| loginName | String | NULL | hz0799 | 用户昵称 |
| avatar | String | NULL | img/avatar-120.png | 用户头像地址 |
| mobileNumber | String | NOT NULL | 13563669636 | 登陆用户电话（唯一） |
| QQNumber | String | NULL | 332308801 | QQ号码 |
| sex | string | NULL | 男 | 用户性别 |
| password | String | NOT NULL | 111111 | 用户密码（必须大于6位） |
| birthday | String | NULL | 2000-01-01 | 用户的生日 |
| userType | Number | NULL | 0 | (根据用户的交易额，给用户不同的级别，获得不同的优惠特权)  0；一级(默认)  1：二级  … |
| location | |  |  | | --- | --- | | Location |  | | NOT NULL | 广东省深圳市南山区XX村XX栋XX号 | 用户当前居住地公开信息。如：location.city获取其中的city数据 |
| created | Date | NOT NULL | 2000-01-01 00:00:00 | 用户注册时间。系统默认当前时间 格式:yyyy-MM-dd HH:mm:ss |
| last\_visit | Date | NOT NULL | 2000-01-01 00:00:00 | 最近登陆时间。系统默认当前时间 格式:yyyy-MM-dd HH:mm:ss |
| status | String | NOT NULL | normal | 状态。可选值:normal(默认正常),inactive(未激活),delete(删除),reeze(冻结),supervise(监管) |

**JSON示例**

1. **根据用户输入正确的用户名和密码获得用户信息对象包含用户地址集合：**

{

" userId ": "10001",

" loginName ": "zaku",

" avatar ": "images/header.png",

" mobileNumber ": "13632336632",

" QQNumber ": "123123123",

" sex ": "男",

" password ": "123456",

" birthday ": "2001-01-06",

" userType ": "一级",

" location ": [

{

" addressId ": "1",

" UserId ": "10001",

" address ": "华星路99号创业大厦",

" city ": "杭州",

" state ": "浙江",

" country ": "中国",

" district ": "西湖区"

},

{

" addressId ": "2",

" UserId ": "10002",

" address ": "深南大道",

" city ": "深圳",

" state ": "广东",

" country ": "中国",

" district ": "南山区"

}

],

" created ": "2001-01-06 10：12",

" last\_visit ": "2001-01-06 10：12",

" status ": "1"

}

1. 根据用户输入的注册信息如：用户名，密码，地址，使用POST请求，把这些参数传入到服务器端，服务器端使用request接受这些参数，调用注册的方法，返回一个服务器结果码如： success成功 error失败
2. 3.用户修改个人信息，如：用户名，密码，地址，使用POST请求，把这些参数传入到服务器端，服务器端使用request接受这些参数，调用注册的方法，返回一个服务器结果码如：success成功 error失败

# 地址API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| addressId | number | NOT NULL | 1 | 地址主键PK |
| UserId | number | NOT NULL | 10001 | 用户编号 FK |
| address | String | NOT NULL | 华星路99号创业大厦 | 详细地址，最大256个字节（128个中文） |
| city | String | NOT NULL | 杭州 | 所在城市（中文名称） |
| state | String | NOT NULL | 浙江 | 所在省份（中文名称） |
| country | String | NOT NULL | 中国（默认） | 国家名称 |
| district | String | NOT NULL | 西湖区 | 区/县（只适用于物流API） |

# 类别API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| TypeId | Number | NOT NULL | 1 | 商品类别编号PK |
| TypeName | String | NOT NULL | 水果 | 产品类型名称 |

**主类型JSON示例**

**2.加载所有的主类型的数据**

{" Type": [

{

" TypeId ": "1001",

" TypeName ": "水果"

},

{

" TypeId ": "1002",

" TypeName ": "日用品"

}

]

}

# 商家API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| SellerId | Number | NOT NULL | 30001 | 商家编号PK |
| SellerName | String | NOT NULL | 美宜佳 | 商家名称 |

{" Seller": [

{

" SellerId ": "1001",

" SellerName ": "百果园"

},

{

" SellerId ": "1002",

" SellerName ": "KFC"

}

]

# 产品API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| ProductID | Number | NOT NULL | 1001 | 所属产品编号 PK |
| ProductName | String | NOT NULL | 苹果 | 产品名称 |
| productPrice | Number | NOT NULL | 13 | 价格 |
| TypeId | Number | NOT NULL | 1003 | 产品主类型编号 FK |
| MeasureUnit | String | NOT NULL | 斤 | 计量单位 |
| productPicturePath | String | NOT NULL | Appler.png | 产品照片地址 |
| productDescription | String | NOT NULL | 好苹果… | 产品描述 |
| ProductStocks | Number | NOT NULL | 300009 | 产品库存 |
| SellerId | Number | NULL | 1 | 商家编号FK |

产品**JSON示例**

**4.加载产品的Json数据:要加一个产品名称的属性**

1. **所有的数据**

{ " Product ":[{

" ProductID": "10001",

" productName": "苹果",

" productPrice": "12",

" TypeId": "1",

" MeasureUnit": "个",

" productPicturePath": " images/apple.png ",

" productDescription": "上好的苹果..",

" ProductStocks": "30000",

" SellerId": "1",

“Evaluation”:[{

" EvaluationID ": "0019 ",

" ProductID ": "0019",

" EvaluationComments ": "很新鲜",

" EvaluationMarks ": "8"

}]

},

{

" ProductID": "10003",

" productName": "咖啡",

" productPrice": "15",

" TypeId": "2",

" MeasureUnit": "杯",

" productPicturePath": " images/coft.png ",

" productDescription": "醇香咖啡..",

" ProductStocks": "30000",

" SellerId": "3"

}]

}

1. **根据商品的名称加载相应的产品json数据**

**参数为：商品的名称，**

**返回值：为一个商品的对象，没有则返回没有数据集合**

{

" ProductID": "10002",

" productName": "香蕉",

" productPrice": "15",

" TypeId": "1",

" MeasureUnit": "支",

" productPicturePath": " images/banana.png ",

" productDescription": "上好的香蕉..",

" ProductStocks": "30000",

" SellerId": "1"

}

1. **根据主类型的编号和商家的编号加载相应的产品json数据**

**参数为：主类型的编号和商家的编号**

**返回值：为一个商品的对象或者集合，没有则返回没有数据集合**

{ " Product ":[{

" ProductID": "10001",

" productName": "苹果",

" productPrice": "12",

" TypeId": "1",

" MeasureUnit": "个",

" productPicturePath": " images/apple.png ",

" productDescription": "上好的苹果..",

" ProductStocks": "30000",

" SellerId": "1"

},

{

" ProductID": "10002",

" productName": "香蕉",

" productPrice": "15",

" TypeId": "1",

" MeasureUnit": "支",

" productPicturePath": " images/banana.png ",

" productDescription": "上好的香蕉..",

" ProductStocks": "30000",

" SellerId": "1"

}]

}

1. **根据商家的编号加载相应的产品json数据**

**参数为：商家的名称，**

**返回值：为一个商品的对象或者集合，没有则返回没有数据集合**

{

" ProductID": "10001",

" productName": "苹果",

" productPrice": "12",

" TypeId": "1",

" MeasureUnit": "个",

" productPicturePath": " images/apple.png ",

" productDescription": "上好的苹果..",

" ProductStocks": "30000",

" SellerId": "1"

}

# 订单API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| order\_id | Number | NOT NULL | 3434 | 订单编号 PK |
| OrderState | Number | NOT NULL | 1 | 订单状态定义==>（  0:等待系统确认 （默认）  1:系统确认  2:配送人员确认  3:正在配送  4:完成购买  ） |
| create\_time | Date | NOT NULL | 2000-01-01 00:00:00 | 订单创建时间 默认系统时间 |
| EvaluationState | String | NOT NULL | 0 | 订单评价状态  0：已评价 1：没评价（默认） |
| Refund\_status | String | NULL | 1 | 退款状态：  0.申请退款  1.退款审核中  2.退款成功  3.退款失败  4.退款关闭 |
| total\_count | Number | NOT NULL | 3 | 总数量 |
| total\_price | Number | NOT NULL | 1200 | 总金额 |
| relation\_Address | String | NOT NULL | 广东省深圳市南山区XX村XX栋XX室 | 联系人地址 |
| relation\_name | String | NOT NULL | 张三 | 联系人姓名 |
| relation\_mobile | String | NOT NULL | 13800138000 | 联系人手机号 |
| relation\_phone\_bak | String | NULL | 13800138000 | 联系人备用电话 |
| relation\_email | String | NULL | xxx@qq.com | 联系人邮箱 |
| DeliverNumber | Number | NULL | 1001 | 配送人员编号FK |
| latestTime | Date | NULL | 2000-01-01 00:00:00 | 预计最晚达到时间 |
| extra | String | NULL | 要冰的可乐 | 备注 |

**订单JSON示例**

1. **根据用户名加载所拥有的订单**

{

" order\_id": "1001",

" OrderState": "1",

" create\_time": "2001-01-06 11：23",

" EvaluationState ": "1",

" Refund\_status": "1",

" total\_count": "9",

" total\_price": "9999",

" relation\_Address": "广东省深圳市南山区XX村XX栋XX室",

" relation\_name": "张某",

" relation\_mobile": "13266996369",

" relation\_phone\_bak ": "13636623321",

" relation\_email": "zhang@163.co m",

" Deliver": [{

" DeliverName ": "XXx",

" DeliverPhone ": "13266996369",

" DeliverID k ": "10001"

}],

" latestTime": "2000-01-01 00:00:00",

" extra": "要冰的可乐",

" priductDetail": [

{

" priductDetailId": "1",

" ProductID": "1001",

" OrderID": "1001",

" MeasureUnit": "杯",

" CommentState": "1",

" productPrice": "23",

" PayMethod": "1",

" PicturePath": "images/apple.png"

},

{

" priductDetailId": "2",

" ProductID": "1002",

" OrderID": "1001",

" MeasureUnit": "斤",

" CommentState": "1",

" productPrice": "20",

" PayMethod": "1",

" PicturePath": "images/xxxx.png"

},

{

" priductDetailId": "3",

" ProductID": "1003",

" OrderID": "1001",

" MeasureUnit": "杯",

" CommentState": "1",

" productPrice": "19",

" PayMethod": "1",

" PicturePath": "images/xxxx.png"

}

]

}

# 订单详情API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| priductDetailId | Number | NOT NULL | 10001 | 订单详情编号 PK |
| ProductID | Number | NOT NULL | 10001 | 产品编号 FK |
| OrderID | Number | NOT NULL | 10001 | 订单编号FK |
| MeasureUnit | String | NOT NULL | 杯 | 产品单位 |
| CommentState | String | NOT NULL | 很好喝的可乐 | 评论状态 |
| productPrice | number | NOT NULL | 8 | 产品单价 |
| PayMethod | String | NOT NULL | 货到付款 | 付款方式：0：货到付款（默认）  1：支付宝  2：银联  3：微信付款 |
| PicturePath | String | NOT NULL | Default/png | 商品图片的绝对地址 |

# 我的收藏API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| CollectId | Number | NOT NULL | 10001 | 收藏编号 PK |
| userId | Number | NOT NULL | 10001 | 用户编号 FK |
| ProductID | Number | NOT NULL | 10001 | 产品编号 FK |

**1.获得收藏的集合**

**参数为：用户的编号**

**返回值：为一个收藏的产品的对象或者集合，没有则返回没有数据集合**

{ " Collect":[{

" CollectId ": "1001",

" userId ": "1002",

" ProductID": "10001",

" productName": "苹果",

" productPrice": "12",

" TypeId": "1",

" MeasureUnit": "个",

" productPicturePath": " images/apple.png ",

" productDescription": "上好的苹果..",

" ProductStocks": "30000",

" SellerId": "1"

},

{

" CollectId ": "1001",

" userId ": "1002",

" ProductID": "10002",

" productName": "香蕉",

" productPrice": "15",

" TypeId": "3",

" MeasureUnit": "支",

" productPicturePath": " images/banana.png ",

" productDescription": "上好的香蕉..",

" ProductStocks": "30000",

" SellerId": "1"

}]

}

2.删除我的收藏

参数：收藏的编号

返回值：用户修改我的收藏，使用POST请求，把这些参数传入到服务器端，服务器端使用request接受这些参数，调用注册的方法，返回一个服务器结果码如：success成功 error失败

# 配送人员信息表API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| DeliverID | Number | NOT NULL | 0001 | 配送人员编号 PK |
| DeliverName | String | NOT NULL | 张三 | 配送人员姓名 |
| DeliverPhone | Number | NOT NULL | 12345678901 | 配送人员电话 |
| DeliverState | Number | NOT NULL | 1 | 配送人员状态 0：空闲（默认） 1：忙碌 2：不当班 |

# 配送信息表API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| dispatchingID | Number | NOT NULL | **1003** | 配送编号 PK |
| OrderID | Number | NOT NULL | 1234 | 订单编号 FK |
| DeliverID | Number | NOT NULL | 1234 | 配送人员编号 FK |
| DeliverInfoId | String | NOT NULL | 姓名+联系电话 | 配送人员信息编号 FK |

JSON示例：

{

" dispatchingID ": "00001",

" OrderID ": "100001",

" DeliverID ": "10009",

" DeliverInfoId ": "小李"

}

# 活动表API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| PromotionID | Number | NOT NULL | **3** | 活动编号 PK |
| productId | Number | NOT NULL | 00001 | 产品编号 FK |
| PromotionType | Number | NOT NULL | **0** | 活动类型 0:秒杀 1：打折 |

JSON示例：

{

" PromotionID ": "00001",

" PromotionType ": "0",

" priduct": [

{

" priductDetailId": "1",

" ProductID": "1001",

" OrderID": "1001",

" MeasureUnit": "杯",

" CommentState": "1",

" productPrice": "23",

" PayMethod": "1",

" PicturePath": "images/apple.png"

}

}

# 产品评价表API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| EvaluationID | Number | NOT NULL | 1005 | 评价编号 PK |
| ProductID | Number | NOT NULL | 1009 | 产品编号 FK |
| EvaluationComments | String | NOT NULL | 很新鲜 | 评价详情 |
| EvaluationMarks | Number | NOT NULL | 8 | 评价分数 |

JSON示例：

{

" EvaluationID ": "0019 ",

" ProductID ": "0019",

" EvaluationComments ": "很新鲜",

" EvaluationMarks ": "8"

}

# 订单评价表API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| OrderEvaluationID | Number | NOT NULL | 0017 | 订单评价编号 PK |
| OrderID | Number | NOT NULL | 0017 | 订单编号 FK |
| DeliverySpeed | Number | NOT NULL | **8** | 派送速度 |
| deliveryManner | Number | NOT NULL | **8** | 服务态度 |
| deliveryPack | Number | NOT NULL | **8** | 外观包装 |
| Remarks | String | NULL | 很快就到了 | 备注 |

JSON示例：

{

"OrderEvaluationID": "0017 ",

"OrderID": "0017",

"DeliverySpeed ": "8",

"deliveryManner ": "8",

"deliveryPack ": "8 ",

"Remarks ": "很快就到了 "

}

# 派送表API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| sendID | Number | NOT NULL | 0001 | 派送编号 PK |
| orderID | Number | NOT NULL | 0001 | 订单编号 FK |
| orderTime | Date | NOT NULL | 2000-01-01 00:00:00 | 下单时间（系统时间，东八区） |
| BuyTime | Date | NULL | 2000-01-01 00:00:00 | 购买时间系统时间，东八区） |
| deliveryTime | Date | NULL | 2000-01-01 00:00:00 | 派送时间系统时间，东八区） |
| OrderDone | Date | NULL | 2000-01-01 00:00:00 | 完成时间系统时间，东八区） |

JSON示例：

{

" sendID ": "0001",

"orderID ": "0001",

" orderTime ": "2001-01-06 10：12",

" BuyTime ": "2001-01-06 10：12",

" deliveryTime ": "2001-01-06 10：12",

" OrderDone ": "2001-01-06 10：12",

}

# 美食教程API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| foodID | Number | NOT NULL | 001 | 美食教程编号 FK |
| food\_image\_url | String | NOT NULL | URL | 美食图片地址 |
| productID | Number | NOT NULL | 00001 | 食材 产品中 FK(字符串形式如：产品编号，编号2，编号3的形式) |
| instruction | string | NULL | 放1/3杯水 | 操作指南 |
| Operation | String | NOT NULL | 是一个特色色拉 | 介绍 |

JSON示例：

{

"foodID": "001",

"food\_image\_url": "http: //tp1.sinaimg.cn/1404376560/50/0/1",

"productID": "00001,00002,00003",

" instruction ": "放1/3杯水",

" Operation ": "是一个特色色拉",

}

# 用户协议API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| agreeID | Number | NOT NULL | 01 | 用户协议编号 FK |
| agreeText | String | NOT NULL | 用户协议 | 用户协议 |

JSON示例：

{

" agreeID ": "01",

" agreeText ": "请仔细阅读用户协议"

}

# 用户使用指南API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| helpID | Number | NOT NULL | 10 | 用户使用指南编号 PK |
| helpText | String | NOT NULL | 用户指南 | 用户使用指南 |

请求参数：

返回结果：

JSON示例：

{

" helpID ": "10",

" helpText ": "请仔细阅读用户使用指南"

}

# 购物车API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否为空** | **示例值** | **描述** |
| CartID | Number | NOT NULL | 1 | 购物车编号 PK |
| ProductID | Number | NOT NULL | 1 | 产品编号 FK |
| userId | Number | NOT NULL | 1 | 用户编号 FK |
| quantity | Number | NOT NULL | 3 | 数量 |
| createDate | Number | NOT NULL | 2014-09-10 10:12 | 创建时间 |

1.获得所有的购物车的信息，用于生成订单

{

" CartID ": "1001",

" userId ": "1",

" quantity ": "2001-01-06 11：23",

" createDate ": "1",

" priductDetail": [

{

" priductDetailId": "1",

" ProductID": "1001",

" MeasureUnit": "杯",

" productPrice": "23",

" PicturePath": "images/apple.png"

},

{

" priductDetailId": "2",

" ProductID": "1002",

" MeasureUnit": "斤",

" productPrice": "20",

" PicturePath": "images/xxxx.png"

},

{

" priductDetailId": "3",

" ProductID": "1003",

" MeasureUnit": "杯",

" productPrice": "19",

" PicturePath": "images/xxxx.png"

}

]

}

2.购物车的添加商品 add

参数：用户编号，产品编号，产品数量，

返回值：服务器响应码success成功 error失败

3.购物车的修改商品数量 update

参数：用户编号，产品编号，产品数量，

返回值：服务器响应码success成功 error失败

4.购物车的删除商品

参数：用户编号，产品编号，产品数量，

返回值：服务器响应码success成功 error失败