# 商城易后端概述文档

目录

[商城易后端概述文档 1](#_Toc9069)

[1. 项目业务功能概述 3](#_Toc27013)

[2.项目架构设计 3](#_Toc22531)

[2.1 总架构 3](#_Toc12454)

[2.2 子模块设计 3](#_Toc18203)

[3.项目功能概述 5](#_Toc15505)

[4.数据库表设计 5](#_Toc9180)

[5.通信方式 8](#_Toc103)

[6. 可调用接口 8](#_Toc14127)

[注意事项 9](#_Toc25228)

[附录 10](#_Toc4738)

[返回代码表 10](#_Toc18669)

# 项目业务功能概述

项目简介

架构

语言/框架

UML类图

数据库设计

接口设计，用途

问题与解答

迭代

历史包袱

关注:

产品文档，原型的理解，技术文档的书写

# 2.商城系统项目架构设计

## 2.1 总架构图

待定

### 2.2 子模块设计-服务商子模块(APP:Service\_Shop)

#### 2.2.1.数据库表设计(service\_shop/models.py):

服务类别表(ServiceCategory)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| name(服务类别) | integer | 50 |  |  |
| depth(深度) | integer |  |  | 服务种类的一级类目 |
| code(类别识别码) | varchar | 50 |  |  |
| service\_area(服务类目) | varchar |  | 默认为邻里服务PRIVATE | 可选:邻里服务/城市服务 |
| parent | 自关联的一对多外键 |  |  | 反查时的关联名称:children |
| order(排序) |  |  |  |  |

服务规格表(ServiceSpecification)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| name(规格名) | varchar |  |  |  |
| media(多对多外键) |  |  |  |  |
| code(规格识别码) | varchar | 512 |  |  |
| cooperate\_price(协议价) | decimal | 20 |  |  |
| suggested\_price(指导价) | decimal | 32 |  |  |
| unit(单位) | varchar |  |  |  |

服务表(Service)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| name(服务名称) | integer |  | √ | √ |
| category(外键关联服务类别表) |  |  |  | 反查时的关联名称为service |
| picture(多对多外键) | varchar | 512 |  |  |
| slug(服务识别码) | varchar | 20 |  |  |
| specifications(多对多外键) |  |  |  |  |
| pay\_method(付费方式) | Int |  |  |  |
| click\_method(操作方式) | Int |  |  |  |
| score(评分) |  |  |  |  |
| community(外键关联到PrivateDomain) |  |  |  | 反查时的关联名称为service |

招聘表(RecruitService)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| name(招聘名称) | integer |  | √ | √ |
| category(外键关联服务类别表) |  |  |  |  |
| picture(多对多外键) |  |  |  |  |
| position(岗位) | varchar | 50 |  |  |
| salary(工资) | varchar | 50 |  |  |

购买服务表(BuyService)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| created\_time |  |  | √ | √ |
| is\_free(是否免费) |  |  |  |  |
| name(服务商名) | varchar | 512 |  |  |
| lifetime(有效时长) | Int | 20 |  |  |
| status(状态) | varchar | 32 |  |  |
| ry(外键关联到融云表) |  |  |  |  |
| service(外键关联服务表) |  |  |  |  |
| category(外键关联服务类别表) |  |  |  |  |
| community(外键关联PrivateDomain) |  |  |  |  |
| city\_info(城市信息) |  |  |  |  |
| group\_id(群组id) |  |  |  |  |

服务价格表ServicePrice

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| service\_category(外键关联服务类别表) |  |  |  |  |
| price(价格) |  |  |  |  |
| cooperate\_price(协议价) |  |  |  |  |
| tags(标签) | varchar | 50 |  |  |

服务订单表ServiceOrder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| order(订单号) | integer |  | √ | √ |
| amount(金额) | integer |  |  |  |
| quantity(购买数量) | varchar | 512 |  |  |
| status(状态) | varchar | 20 |  |  |
| service(外键关联服务表) |  |  |  | 反查时关联名称service\_order\_of |
| category(外键关联服务类别表) |  |  |  | 反查时关联名称service\_order\_of |
| spec(外键关联服务规格表) |  |  |  | 反查时关联名称buy\_service\_of |
| is\_vip(是否为vip下单) |  |  |  |  |
| user(外键关联到User) |  |  |  | 反查时关联名称service\_order\_of |
| community(外键) |  |  |  | 反查时关联名称service\_order\_of |
| split\_bill(分账单号) |  |  |  |  |
| dispatch\_order\_id(派单易订单号) |  |  |  |  |
| transaction(流水单号) |  |  |  |  |
| longitude(经度) |  |  |  |  |
| latitude(纬度) |  |  |  |  |
| shipping\_address\_json(收货地址) |  |  |  |  |
| extra(额外信息) |  |  |  |  |
| created\_time(创建时间) |  |  |  |  |

### 2.2.2可调用接口(service\_shop/url.py)

| **名称** | **描述** |
| --- | --- |
| api/login | 登录接口 |
| api/v1/buy\_service | 购买服务接口 |
| add-service-order | 增加服务单 |
| api/v1/query\_service\_item | 查询服务清单接口 |
| api/v1/query\_service\_shop | 查询报价 |
| api/v1/query\_service\_shop\_item | 查询子类目 |
| api/v1/submit\_order | 提交订单 |
| api/v1/query\_order | 查询订单接口 |
| api/v1/query\_specification | 查询规格接口 |
| api/v1/comfirm\_order | 订单付款成功接口 |
| api/v1/cancel\_order | 取消订单接口 |
| api/v1/change\_order\_status | 更改订单状态接口 |
| api/v1/query\_self\_order | 查询自己下过的所有服务单 |
| api/v1/confirm\_service | 确认服务完成接口 |
| api/v1/query\_order\_detail/<str:order> | 查询订单详情接口 |

### 2.2.3.接口详情描述(service\_shop/new\_views.py)

本接口基于 HTTP GET/POST方式的通信方式，请求和返回数据是JSON，使用UTF-8编码

###### 1.查询服务清单接口

请求方式:GET请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/query\_service\_item

请求说明:查询服务类别表ServiceCategory,返回数据为服务类别表中对应字段的全部数据

示例如下：



###### 2.查询报价接口

请求方式:GET请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/query\_service\_shop

请求说明:查询服务类别表ServiceCategory获取某个服务的报价

请求传参举例:

{

“community”: “(非必要)”,

“service\_area”:”public(城市服务)”,

“depth”:”2(二级类目)”,

“parent”:”public\_67df540b69(鲜花绿植类目的类别识别码)”

}

返回数据示例:(鲜花绿植类目下所有子服务类别的报价)



###### 3.查询子类目接口

请求方式:GET请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/query\_service\_shop\_item

请求内容:查询服务表中的

请求传参与返回数据举例:

{“community”: “xxx”, “category”:”public”}

返回数据:

###### 4.提交订单接口

请求方式:POST请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/submit\_order

请求传参:

{“code”:”xxx”,“shipping\_address”:”xxx”,“quantity”:”xxx”,”spec”:”xxx”,”is\_vip”:”xx”,”community”:”xx”}

查询逻辑:

①根据code参数查询服务类别表得到service对象

②根据shipping\_address参数查询用户地址表

③根据service对象查询服务价格表得到service\_price对象

④根据spec参数查询服务规格表

⑤

⑥创建服务订单对象(服务订单表中的一行数据)

###### 5.查询订单接口

请求方式:GET请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/query\_order

请求参数:{“token”:”xx”, “order\_id”: “xx”}

###### 6.查询规格接口(查询某个社区对应的某个服务类别下的所有服务规格描述)

请求方式:GET请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/query\_specification

请求参数:{

“community”:”96(青羊宫小区)”,

“category”: “private\_c9b0614c01\_b47af172e3(物业维修服务的类别识别码)”

}

返回数据示例:返回服务规格表中对应字段的数据



**订单付款成功接口**

**取消订单接口**

**更改订单状态接口**

请求方式:POST请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/change\_order\_status

请求传参:{"order": “订单号”, "status": “订单状态”}

请求说明:根据订单号查询服务订单表修改订单状态

**查询自己下过的所有服务单的接口**

请求方式:GET请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/query\_self\_order

**确认服务完成接口**

**查询订单详情接口**

请求方式:GET请求

请求链接:http://api.esoaru.net/api-lhb/service-shop/api/v1/query\_order\_detail/<str:order>

请求说明:根据订单号查询服务订单表(service\_shop\_serviceorder)返回对应订单的全部数据

返回数据示例:



### 2.3 子模块设计-商品模块

#### 2.3.1.数据库表设计(shop\_manager/models.py):

商品审核表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| user(操作人-外键关联用户表) |  |  |  |  |
| opertype(操作类型) |  |  |  |  |
| opercontent(操作内容) |  |  |  |  |
| product(商品-外键关联产品表) |  |  |  |  |
| created(创建时间) |  |  |  |  |
| updated(更新时间) |  |  |  |  |
| status(审核状态) |  |  |  |  |

### 产品表Product(属于catalogue应用的)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| structure(产品结构) |  |  |  | （独立产品,母产品,子产品） |
| is\_public |  |  |  |  |
| upc(产品标识符) |  |  |  |  |
| parent(自关联外键) |  |  |  | 反查时关联名称children |
| title(产品标题) |  |  |  |  |
| slug(条形码) |  |  |  |  |
| Description(产品描述) |  |  |  |  |
| meta\_title(元标题) |  |  |  |  |
| meta\_description |  |  |  |  |
| product\_class(外键关联产品类型表) | 一对多外键 |  |  | 反查时关联名称products |
| attributes(外键关联产品属性表) | 多对多外键 |  |  |  |
| product\_options(多对多关联) | 多对多外键 |  |  |  |
| recommended\_products() | 多对多外键 |  |  |  |
| rating |  |  |  |  |
| date\_created |  |  |  |  |
| date\_updated |  |  |  |  |
| categories |  |  |  |  |
| is\_discountable |  |  |  |  |
| order\_by(排序) |  |  |  |  |
| is\_business(是否云仓私域商品) |  |  |  |  |
| pull\_off\_shelves(是否下架商品) |  |  |  |  |
| view\_count(浏览量) |  |  |  |  |

### 订单表Order(属于catalogue应用的)

### 2.3.2可调用接口(shop\_manager/url.py)

| **名称** | **描述** |
| --- | --- |
| api/login | 登录接口 |
| api/logout | 注销接口 |
| api/currentUser | 查询当前用户接口 |
| api/query-product | 查询产品接口 |
| api/query-order | 查询订单接口 |
| api/add-product | 添加单个商品接口 |
| api/add-logistics | 添加物流信息接口 |
| api/add-product-list | 批量上传商品接口 |
| api/confirm-order | 确认订单接口 |
| api/cancel-order | 取消订单接口 |
| api/cancel-order-callback | 取消订单回调接口 |
| api/merchant-delay-delivery | 商户延迟发货接口 |
| api/query-order-detail | 查询单个订单接口 |
| api/query-all-partner | 查询所有合作伙伴接口 |
| api/change-product-status | 下架商品接口 |
| query-user-is-partner/<phone> | 查询用户是否为商户接口 |
| query-total-sales | 查询总销售额接口 |
| query-product-sales | 查询特定商品的咨询数/下单数/销售额/详情页浏览量 |

### 2.3.3.接口详情描述(shop\_manager/views.py)

#### 1.查询产品接口

请求方式:GET

请求链接:http://api.esoaru.net/api-lhb/shop-manager/api/query-product

返回数据示例



#### 查询订单接口

### 2.2 子模块设计-供货商模块

### 2.4 子模块-二手商城(APP: Second\_Hand)

#### 2.4.1.数据库表设计(second\_hand/models.py):

二手商品表(模型类:SecondHand)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| title |  |  |  |  |
| original\_price(原价) |  |  |  |  |
| second\_hand\_price(二手价) |  |  |  |  |
| user | 一对多外键 |  | 外键关联扩展用户表 | 反向查询关联名称:second\_hand |
| created(创建时间) |  |  |  |  |
| updated(更新时间) |  |  |  |  |
| status(状态) |  |  |  |  |
| Content |  |  |  |  |
| Community | 一对多外键 |  |  | 反向查询关联名称:second\_hands |
|  |  |  |  |  |

二手图片表(模型类:SecondHandImage)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| product | 一对多外键 |  |  | 反向查询关联名称:images |
| url |  |  |  |  |
| sort |  |  |  |  |
| created(创建时间) |  |  |  |  |
| updated(更新时间) |  |  |  |  |

二手订单表(SecondHandOrder)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| order |  |  |  |  |
| product |  |  |  |  |
| amount |  |  |  |  |
| user | 一对多外键 |  | 外键关联扩展用户表 | 反向查询关联名称:second\_hand\_orders |
| created(创建时间) |  |  |  |  |
| shipping\_address\_json |  |  |  |  |
| satus |  |  |  | 7种可选状态,默认为待支付状态 |

### 2.4.3.接口详情描述(second\_hand/views.py)

lhb.public.private\_domain\_buyproductryid

lhb.public.catalogue\_product

### 2.5 小区(APP: Private\_domain)

### 2.5.2 utils文件配置介绍(utils/api\_product)

生成产品接口(gen\_products)

### 2.5.3.接口详情描述(private\_domain\_api\_views.py)

查询小区热门商品 / 高额营销费商品 / 全部可首单购买的商品

请求URL:http://api.esoaru.net/api-lhb/private\_domain\_api/query-selection-products

请求方式:GET

|  |  |
| --- | --- |
| page | 非必要 |
| community | 必要(社群号) |
|  |  |

查询逻辑:(需要熟悉构造Q查询的逻辑)

①根据参数community查询私域表PrivateDomain获得该表中一行记录

②根据第①步中获取的对象查询小区选品官表(selection\_officer)获得该表中的一行记录

③根据第②步中获得的记录查询(private\_domain\_buyproductryid)获取被购买过的商品记录

④根据第③步获得的记录查询产品表(catalogue\_product)中的非下架的产品

⑤第④步获取到的产品是个queryset对象,将其序列化返回

订单表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| transfer\_order\_no |  |  |  | 转账单号 |
| community | 一对多外键 |  |  | 小区订单 |
| split\_bill |  |  |  | 分账单号 |
| group\_id |  |  |  | 发红包的群组id |
| buy\_member\_card |  |  |  | 购买体验卷 |
| open\_consult |  |  |  | 开放咨询 |
| open\_selection\_officer |  |  |  | 开通小区选品官 |
| extra\_info |  |  |  | 订单额外信息 |
| inviter\_user |  |  |  | 通过链接购买的 链接邀请人 |

产品分账自定义规则表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列 | 数据类型 | 长度/精度 | 不为NULL | 说明 |
| source |  |  |  | 分账源(三种分账关系)APP/THRID/XQPP推广关系 |
| product | 一对多外键 |  |  | 小区订单 |
| prorate |  |  |  | 分账比例 |
| created\_at |  |  |  | 创建时间 |
| updated\_at |  |  |  | 更新时间 |

# 注意事项

# 附录

## 返回代码表