

# 天津大学

## 诚园小商城的设计文档



学 院	智能与计算学部
专 业	软件工程
年 级	2018级
姓 名	李昭卓
学 号	3018216132
姓 名	吕鸿江
学 号	3018216134
姓 名	康子文
学 号	3018216131
姓 名	马力
学 号	3018216137

2020 年 9 月 18 日

# 目 录

第一章	业务逻辑设计	1
1.1	业务逻辑设计:	1
1.2	总体目录结构图:	1
第二章	接口设计	3
2.1	首页 /home	3
2.2	商品分类 /categories	5
2.3	商品 /good	6
2.4	订单 /order	8
2.5	用户 /user	11
第三章	数据库设计	14
3.1	E-R图	14
3.2	商品表 good:	14
3.3	用户表 user:	15
3.4	一级分类表cat1:	15
3.5	二级分类表cat2:	15
3.6	订单表form:	15

## 第一章 业务逻辑设计

### 1.1 业务逻辑设计：

业务逻辑设计：MVC三层架构编写，上层Controller层负责连接外部接口与下层的service接口，中层service层负责给上层提供功能接口，提供json串形式的数据，调用数据库的相应功能实现相应的功能并封装成json形式的数据，Dao层的接口给service层提供数据接口，Entity层直接操作数据库实现对数据库的基本操作。

数据库在数据库设计，数据层Entity层给上层的service提供8个dao接口，每一个表对应一个dao接口，每一个接口对应一个Entity实体类负责直接实现对数据库执行相应的功能。

### 1.2 总体目录结构图：

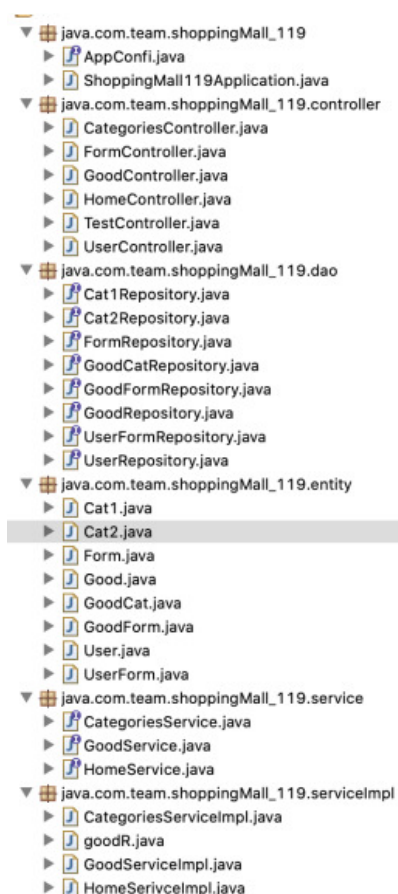


图 1-1

数据库中的8个表对应entity中的8个实体类，可以实现基础的数据库的操作，查询某个属性值或是设置某个属性的值到数据库，这部分直接与数据库相连。dao层的接口与entity相对应，负责给上层的service层提供相应的接口来实现相应的功能。service负责实现各个页面的接口的功能，并将处理的数据封装成json串的形式返回给上层的controller层，controller负责接收来自小程序前端的url的数据请求并调用service层的接口得到相关的json数据并返回给小程序前端。 entity如下：

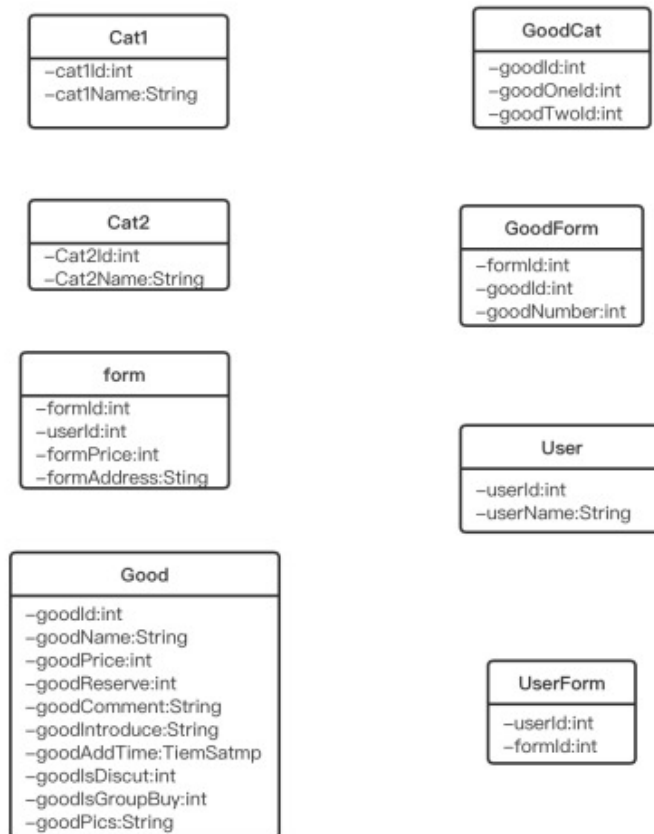


图 1-2

## 第二章 接口设计

### 2.1 首页 /home

#### 2.1.1 轮播图 /home/swiperdata get方法

返回示例：

返回示例：

```
{
  "message":[
    {
      "image_src":"轮播图片地址",
      "good_id":"商品id"
    },
    {},
    {}
  ],
  "meta":{
    "msg":"获取成功",
    "status":200
  }
}
```

图 2-1

#### 2.1.2 轮播图 /home/activity get方法

返回示例：

```
{
  "message":[
    {
      "image_src":"活动图标地址",
      "good":[
        {
          "good_id":"商品id",
          "good_name":"商品名称",
          "good_price":"商品价格",
          "image_src":"商品图片地址"
        },
        {},
        {}
      ]
    },
    {},
    {}
  ],
  "meta":{
    "msg":"获取成功",
    "status":200
  }
}
```

图 2-2

### 2.1.3 轮播图 /home/show get方法

返回示例：

```
{
  "message":[
    {
      "good_id":"商品id"
      "good_name":"商品名称"
      "good_price":"商品价格",
      "image_src":"商品图片地址"
    },
    {},
    {}
  ],
  "meta":{
    "msg":"获取成功",
    "status":200
  }
}
```

图 2-3

## 2.2 商品分类 /categories

### 2.2.1 获取分类 /categories get方法

返回示例：

```
{
  "message":[
    {
      "cat1_id":"一级分类id",
      "cat1_name":"一级分类名称",
      "cat2":[
        {
          "cat2_id":"二级分类id",
          "cat2_name":"二级分类名称",
          "cat2_pic_src":"二级分类图片地址"
        },
        {},
        {}
      ]
    },
    {},
    {}
  ],
  "meta":{
    "msg":"获取成功",
    "status":200
  }
}
```

图 2-4

## 2.3 商品 /good

### 2.3.1 商品搜索 /good/search get方法

参数列表：



参数名	必须	类型	说明
keyword	否	string	搜索关键字
cat1_id	否	number	一级分类
cat2_id	否	number	二级分类
is_discut	否	boolean	是否折扣
is_groupbuy	否	boolean	是否拼团
is_new	否	boolean	是否上新

图 2-5

返回示例：

```
{
  "message":[
    {
      "good_id":"商品id",
      "good_name":"商品名称",
      "good_price":"商品价格",
      "image_src":"商品图片地址"
    },
    {},
    {}
  ],
  "meta":{
    "msg":"获取成功",
    "status":200
  }
}
```

图 2-6

### 2.3.2 商品详情 /good/detail get方法

参数列表：

参数名	是否必须	类型	说明
goods_id	否	int	商品id

图 2-7

返回示例：

```
{
  "message":{
    "good_id": "商品id",
    "good_name": "商品名",
    "good_price": "商品价格",
    "good_number": "商品库存",
    "good_introduce": "富文本",
    "cat1_id": "一级分类id",
    "cat2_id": "二级分类id",
    "is_discut": "true/false",
    "is_groupbuy": "true/false",
    "pics":[
      {
        "image_src": "商品轮播图地址"
      },
      {},
      {}
    ]
  },
  "meta":{
    "msg": "获取成功",
    "status": 200
  }
}
```

图 2-8

## 2.4 订单 /order

### 2.4.1 订单创建 /form/create get方法

参数列表：

参数名	必选	类型	说明
form_date	是	string	订单日期
form_price	是	number	订单总金额
form_num	是	number	订单总数量
address	是	string	收货地址
goods	是	Array	商品数组
nick_name	是	string	用户名

图 2-9

goods字段说明：

参数名	必选	类型	说明
good_id	是	number	商品id
good_num	是	number	商品数量
good_price		number	单价

图 2-10

返回示例：

返回示例：

```
{
  "message":{
    "state":1
  }
  "meta":{
    "msg":"获取成功",
    "status":200
  }
}
```

图 2-11

## 2.4.2 订单获取 /form/detail get方法

参数列表:

参数名	必选	类型	说明
state	是	number	订单状态
nick_name	是	string	用户昵称

图 2-12

state状态说明:

state参数值	含义
0	全部
1	待审核
2	待发货
3	配送中
4	已送达
5	已完成

图 2-13

返回示例:

```
{
  "message": [
    {
      "form_id": "订单id",
      "form_date": "订单日期",
      "form_price": "订单总金额",
      "form_num": "订单总数量",
      "form_state": "订单状态"
    },
    {
      "goods": [
        {
          "good_id": "商品id",
          "goods_num": "商品数量",
          "goods_price": "商品价格"
        },
        {}
      ]
    },
    {}
  ],
  "meta": {
    "msg": "获取成功",
    "status": 200
  }
}
```

图 2-14

## 2.5 用户 /user

### 2.5.1 用户信息 /user/detail get方法

参数列表：

参数名	必选	类型	说明
nick_name	是	string	用户昵称
money	是	num	用户余额

图 2-15

返回示例：

```
{  
  "message": [  
    {  
      "state": 1  
    }  
  ],  
  "meta": {  
    "msg": "获取成功",  
    "status": 200  
  }  
}
```

图 2-16

### 2.5.2 用户余额 /user/money get方法

参数列表：

参数名	必选	类型	说明
nick_name	是	string	用户昵称

图 2-17

返回示例：

```
{
  "message":[
    {
      "money":5000
    }
  ],
  "meta":{
    "msg":"获取成功",
    "status":200
  }
}
```

图 2-18

## 第三章 数据库设计

### 3.1 E-R图

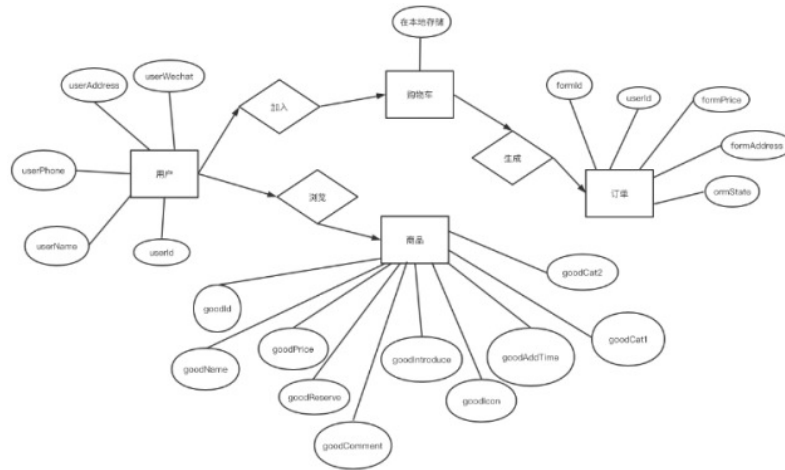


图 3-1

### 3.2 商品表 good:

列名	类型	属性
<u>goodId</u>	int	not null primary key auto_increment
<u>goodName</u>	Varchar(255)	not null
<u>goodPrice</u>	int	not null
<u>goodReserve</u>	int	not null default value:999
<u>goodComment</u>	varchar(255)	default value:"暂无评论"
<u>goodIntroduce</u>	varchar(255)-> 商品的介绍文字 和图片 (富文本)	not null
<u>goodCon</u>	varchar(255)	not null
<u>goodAddTime</u>	time	not null
<u>goodCat1</u>	int(分类的 id)	not null
<u>goodCat2</u>	int( 分类的 二级 id)	not null

图 3-2



### 3.3 用户表 user:

列名	类型	属性
userId	int	not null primary key auto_increment
userName	varchar(255)	not null
userPhone	char(11)	not null
userAddress	varchar(255)	not null
userWechat	varchar(255)	not null

图 3-3

### 3.4 一级分类表cat1:

列名	类型	属性
cat1Id	int	not null primary key auto_increment
cat1Name	varchar (255)	not null

图 3-4

### 3.5 二级分类表cat2:

列名	类型	属性
cat2Id	int	not null primary key auto_increment
cat2Name	varchar (255)	not null

图 3-5

### 3.6 订单表form:

列名	类型	属性
formId	int	not null primary key auto_increment
userId	int	not null
formPrice	int	not null
formAddress	varchar(255)	not null
formState	int	not null default value:1

图 3-6

orderState: 待审核 1、待发货 2、配送中 3、已签收 4、确认订单完成 5。

购物车内容存在本地（不需要后端）

Default value