

# 谷粒商城

商品服务



## 一、基础概念

### 1、三级分类



### 2、SPU 与 SKU

**SPU: Standard Product Unit (标准化产品单元)**

是商品信息聚合的最小单位，是一组可复用、易检索的标准化信息的集合，该集合描述了一个产品的特性。





iPhone X 是 SPU、MI 8 是 SPU

iPhone X 64G 黑曜石 是 SKU

MI 8 8+64G+黑色 是 SKU

#### SKU: Stock Keeping Unit (库存量单位)

即库存进出计量的基本单元，可以是以件，盒，托盘等单位。SKU 这是对于大型连锁超市 DC（配送中心）物流管理的一个必要的方法。现在已经被引申为产品统一编号的简称，每种产品均对应唯一的 SKU 号。

### 3、基本属性【规格参数】与销售属性

每个分类下的商品共享规格参数，与销售属性。只是有些商品不一定要用这个分类下全部的属性：

- 属性是以三级分类组织起来的
- 规格参数中有些是可以提供检索的
- 规格参数也是基本属性，他们具有自己的分组
- 属性的分组也是以三级分类组织起来的
- 属性名确定的，但是值是每一个商品不同来决定的

## 二、接口编写

### 1、HTTP 请求模板

```
"http-get 请求": {
  "prefix": "httpget",
  "body": [
    "this.\\$http{",
    "url: this.\\$http.adornUrl(",
    "method: 'get'",
    "params: this.\\$http.adornParams({})",
    "}).then(({data}) => {",
    "}"
  ],
  "description": "httpGET 请求"
}

"http-post 请求": {
  "prefix": "httppost",
  "body": [
    "this.\\$http{",
    "url: this.\\$http.adornUrl(",
    "method: 'post'",
    "data: this.\\$http.adornData(data, false)",
    "}).then(({ data }) => { });"
  ],
  "description": "httpPOST 请求"
}
```

### 2、JSR303 数据校验

#### 1)、使用步骤

- 标注校验注解

javax.validation.constraints 中定义了非常多的校验注解

@Email、@Future、@NotBlank、@Size 等

- 使用校验功能

@Valid 开启校验功能

- 提取校验错误信息

BindingResult 获取校验结果

- 分组校验与自定义校验

Groups 定义校验分组信息；

可以编写自定义校验注解和自定义校验器

默认情况下，异常信息会从应用的 classpath 下的 ValidationMessages.properties 文件中加载；

### 3、全局异常处理

@ControllerAdvice+@ExceptionHandler

系统错误码

```
/**
 * 错误码和错误信息定义类
 * 1. 错误码定义规则为 5 为数字
 * 2. 前两位表示业务场景，最后三位表示错误码。例如：100001。10:通用 001:系统未知异常
 * 3. 维护错误码后需要维护错误描述，将他们定义为枚举形式
 * 错误码列表：
 * 10: 通用
 * 001: 参数格式校验
 * 11: 商品
 * 12: 订单
 * 13: 购物车
 * 14: 物流
 *
 */
```

### 4、接口文档地址

<https://easydoc.xyz/#/s/78237135>

## 5、Object 划分

### 1.PO(persistent object) 持久对象

PO 就是对应数据库中某个表中的一条记录，多个记录可以用 PO 的集合。PO 中应该不包含任何对数据库的操作。

### 2.DO (Domain Object) 领域对象

就是从现实世界中抽象出来的有形或无形的业务实体。

### 3.TO(Transfer Object) ， 数据传输对象

不同的应用程序之间传输的对象

### 4.DTO (Data Transfer Object) 数据传输对象

这个概念来源于 J2EE 的设计模式，原来的目的是为了 EJB 的分布式应用提供粗粒度的数据实体，以减少分布式调用的次数，从而提高分布式调用的性能和降低网络负载，但在这里，泛指用于展示层与服务层之间的数据传输对象。

### 5.VO(value object) 值对象

通常用于业务层之间的数据传递，和 PO 一样也是仅仅包含数据而已。但应是抽象出的业务对象，可以和表对应，也可以不，这根据业务的需要。用 new 关键字创建，由 GC 回收的。

View object: 视图对象;

接受页面传递来的数据，封装对象

将业务处理完成的对象，封装成页面要用的数据

### 6.BO(business object) 业务对象

从业务模型的角度看，见 UML 元件领域模型中的领域对象。封装业务逻辑的 java 对象，通过调用 DAO 方法，结合 PO,VO 进行业务操作。business object: 业务对象 主要作用是把业务逻辑封装为一个对象。这个对象可以包括一个或多个其它的对象。比如一个简历，有教育经历、工作经历、社会关系等等。我们可以把教育经历对应一个 PO，工作经历对应一个 PO，社会关系对应一个 PO。建立一个对应简历的 BO 对象处理简历，每个 BO 包含这些 PO。这样处理业务逻辑时，我们就可以针对 BO 去处理。

## 7.POJO(plain ordinary java object) 简单无规则 java 对象

传统意义的 java 对象。就是说在一些 Object/Relation Mapping 工具中，能够做到维护数据库表记录的 `persisent object` 完全是一个符合 Java Bean 规范的纯 Java 对象，没有增加别的属性和方法。我的理解就是最基本的 java Bean，只有属性字段及 `setter` 和 `getter` 方法！。

POJO 是 DO/DTO/BO/VO 的统称。

## 8.DAO(data access object) 数据访问对象

是一个 sun 的一个标准 j2ee 设计模式，这个模式中有个接口就是 DAO，它负持久层的操作。为业务层提供接口。此对象用于访问数据库。通常和 PO 结合使用，DAO 中包含了各种数据库的操作方法。通过它的方法，结合 PO 对数据库进行相关的操作。夹在业务逻辑与数据库资源中间。配合 VO，提供数据库的 CRUD 操作。

