

- 商品模块API
 - 数据库表设计
 - 接口说明
 - 1. 获取所有分类名
 - 2. 获取所有商品
 - 3. 获取商品详情
 - 4. 搜索商品

商品模块API

数据库表设计

1. 商品表CommodityInfo

字段名	类型	作用	备注1	备注2
id	int	自增主键	无	无
name	char(180)	商品名	不可为空	无
shelves	boolean	是否在售	默认' True'	无
price	decimal(6,2)	商品价格	不可为空	最多9999.99,保留小数点后2位数
description	text	商品描述	无	无
images	ImageField	商品图片	无	无
classify	ForeignKey	外键	关联CommodityClassify	无

2. 分类表CommodityClassify

字段名	类型	作用	备注1	备注2
id	int	主键		
name	char(150)	分类名字	不可为空	无
description	text	分类描述	无	无

接口说明

1. 获取所有分类名

- 请求URL：<http://127.0.0.1:8000/commodity/classify>
- 请求方式：GET
- 请求格式：无
- 响应格式：

字段名	类型	描述	备注
code	int	200表示请求成功	/
data		返回的所有数据	/
name	string	分类名	/
description	string	分类描述	/

响应示例：

```
json { "code": 200, "data": { "name": '飞鞋', "description": '穿上会飞哟' } }
```

- 异常码：

字段名	类型	描述	备注

2. 获取所有商品

- 请求URL：[<?type=xxx>](http://127.0.0.1:8000/commodity)
- 请求方式：GET
- 请求格式：如果请求url中有查询参数type，则表示请求的是该type下的所有商品；否则，请求的是所有商品。
- 响应格式：

字段名	类型	描述	备注
code	int	200表示请求成功	/
data		返回的所有数据	/
id	int	商品表中的id字段	
name	string	商品名称	/
price	float	商品价格	/

<i>image</i>	string	图片地址	/
--------------	--------	------	---

响应示例：

```
json { "code": 200, "data": { "id": 1001, "name": '飞鞋', "price": 700.89, "image": 'flyshoes.png' } }
```

- 异常码：

字段名	类型	描述	备注

3. 获取商品详情

- 请求URL：<http://127.0.0.1:8000/commodity?id=xxx>
- 请求方式：GET
- 请求格式：url查询参数为商品的id
- 响应格式：

字段名	类型	描述	备注
code	int	200表示请求成功	/
data		返回的所有数据	/
<i>id</i>	int	商品表中的id字段	
<i>name</i>	string	分类名	/
<i>description</i>	string	分类描述	/
<i>shelves</i>	boolean	是否在售	/
<i>price</i>	float	价格	/
<i>image</i>	string	图片地址	/

响应示例：

```
json { "code": 200, "data": { "id": 1001, "name": '飞鞋', "description": '穿上会飞哟', "shelves": true, "price": 700.89, "image": 'flyshoes.png' } }
```

- 异常码：

字段名	类型	描述	备注

4.搜索商品

url:<http://127.0.0.1:8000/commodity/search>

请求方式：POST

请求格式

| 字段名 | 类型 | 描述 | 备注 |

-|-|-

|key|char|关键字|

响应格式:

```
{ 'code': 200, 'data': { 'id': goods.id, 'name': goods.name, 'shelves': goods.shelves, 'price': str(goods.price), 'description': goods.description, 'image': str(goods.images)}}
```