

# 数据库

用户 (id, 用户名, 密码, 资料, 余额, 角色【系统用户1, 管理员2, 游客3】)

地址 (id, 用户名, 地址, 是否默认)

商品 (id, 商品名称, 商品类别, 商品图片, 商品详情, 状态【上架, 下架】, 销量)

类别 (id, 类别【大类/小类/细分】) (极大可能取消,合并到商品表)

订单 (id, 用户名, 商品id, 状态【1-未付款, 2-已付款, 3-已发货, 4-交易成功, 5-交易关闭, 6-已取消】, 开始时间, 结束时间)

购物车 (id, 用户名, 商品id, 数量)

mysql用户名root密码action123

# 前台功能

## 用户登录 POST /user/login

request

字段	类型	解释
username	string	
password	string	

response

errno	int	错误码, 0表示无错误
data	User	用户信息

## 注册 POST /user/register

request

username	string	
password	string	
checkpass	string	
detail	string	详细信息
role	int	1系统用户2管理员3游客

response

errno	int	错误码, 0表示无错误
error	string	错误描述

## 个人资料 GET /user/get

request

uid	int	用户id

response

errno	int	错误码, 0表示无错误
error	string	错误描述
data	UserAndAddr	用户信

## PS. 自定义的数据结构

user的结构

```
{
  "id": 1,
  "name": "dzy",
  "pass": "123",
  "detail": "hahahahaha",
  "balance": 100.0,
  "role": 1
}
```

UserAndAddr的结构

```
{
  "id": 1,
  "name": "dzy",
  "pass": "123",
  "detail": "hahahahaha",
  "balance": 100.0,
  "role": 1,
  "addr": [
    {
      "isDefault": true,
      "place": "xxx省 xxx市"
    },
    {
      "isDefault": true,
      "place": "xxx省 xxx市"
    }
  ]
}
```

## 商品展示

## 热销商品展示 GET /product/list

request

pageSize	int	页大小
pageNum	int	第几页

response

total	int	商品总数
pageSize	int	页大小
pageNum	int	第几页
productList	Product数组	该页的商品

## 分类商品展示 GET /product/list/type

request

productType	string	商品类型
pageSize	int	页大小
pageNum	int	第几页

response

total	int	商品总数
pageSize	int	页大小
pageNum	int	第几页
productList	Product数组	该页的商品

## 商品详情 GET /product/get

request

productId	int	商品id

response

--	--	--

product	Product	商品对象

## 辅助接口

### 获取所有商品分类 GET /product/types

request

不需要参数		

response

typeList	string数组	所有的分类

## PS. 自定义的数据结构

Product

```
{
  "id": 3,
  "name": "name4",
  "ptype": "type2",
  "pic": "1.png, 2.png",
  "detail": "detail",
  "price": 3.0,
  "status": 1,
  "salecnt": 1
}
```

## 购物车

### 把商品添加到购物车 POST

request

username	string	用户名
pid	int	商品id
cnt	int	商品数量

response

errno	int	0成功1失败

## 编辑购物车 POST

request

idList	string	选中要删除的商品的id，用逗号隔开
username	string	用户名

response

errno	int	0成功1失败

## 结算 POST

request

username	string	用户名
idList	string	选中商品的id，用逗号隔开

response

errno	int	0成功1失败

## 列出购物车内所有商品 GET

request

username	string	用户名

response 示例

```
[
  {
    "id": 1,    // 购物车项id
    "count": 2,
    "product": {
      "id": 3,
      "name": "name4",
      "ptype": "type2",
      "pic": "1.png, 2.png",
      "detail": "detail",
      "price": 3.0,
      "status": 1,
      "salecnt": 1
    }
  },
  {
    "id": 2,
    "count": 1,
    "product": {
```

```
        "id": 4,
        "name": "name5",
        "ptype": "type2",
        "pic": "1.png, 2.png",
        "detail": "detail",
        "price": 3.0,
        "status": 1,
        "salecnt": 1
    }
}
```

# 订单

## 订单列表 GET

request

username	string	用户名

response

orders	Order数组	订单对象数组

## 订单详情 GET

request

orderId	订单id	

response

order	Order	订单对象

## PS. 自定义的数据结构

Order

```
// id, 用户名, 商品id, 状态【1-未付款, 2-已付款, 3-已取消】, 开始时间, 结束时间
// 稍后完善
```

# 地址管理

## 列出用户的所有地址 GET

request

uid	int	用户id
addr	string	新增的地址

response

errno	int	0成功1失败

## 增加地址 POST

---

request

uid	int	用户id
addr	string	新增的地址

response

errno	int	0成功1失败

## 删除地址 POST

---

request

aid	int	地址id

response

errno	int	0成功1失败

## 修改地址 POST

---

request

aid	int	地址id
addr	string	新地址

response

errno	int	0成功1失败

## 设置默认地址 POST

request

aid	int	地址id

response

errno	int	0成功1失败

## 支付 POST

request

productId	int	商品id
uid	int	用户id

response

errno	int	0成功1失败

## 充值 POST

request

money	double	充值金额
uid	int	用户id

response

errno	int	0成功1失败

## 商品管理

### 查询所有商品 GET

request

--	--	--



pageSize	int	页大小
pageNum	int	第几页

response

total	int	商品总数
pageSize	int	页大小
pageNum	int	第几页
productList	Product数组	该页的商品

## 查询商品 GET

同 [商品展示 -> 商品详情 GET]

## 新增商品 POST

request

pageSize	int	页大小
pageNum	int	第几页

response

total	int	商品总数
pageSize	int	页大小
pageNum	int	第几页
productList	Product数组	该页的商品

## 修改商品 POST

request

product	Product	商品对象,其中的id属性为要修改的商品id

response

errno	int	0成功1失败

## 上下架商品 POST

request

productId	int	商品id
stat	int	

response

errno	int	0成功1失败