接口文档

一、配置

- **JDK**：JDK22
- **SpringBoot**：SpringBoot3
- **Vue**：Vue3
- **DataBase**：MySQL

二、模块划分详细版（接口文档准备用）

1. 用户模块（user）

- 用户注册
  - 填写用户名、手机号、密码，提交注册
- 用户登录
  - 输入用户名密码登录
- 用户登出
  - 主动退出登录（清除token）
- 获取当前用户信息
  - 获取个人资料（昵称、手机号、头像等）
- 修改用户个人信息
  - 修改头像
  - 修改昵称
  - 修改手机号
  - 修改密码
- 订单记录
  - 查看已完成订单
  - 查看未完成订单
- 用户收藏商品

    - 收藏一个商品
    - 取消收藏一个商品
- 用户收藏店铺
    - 收藏一个店铺
    - 取消收藏一个店铺
- 用户查看收藏列表
    - 获取收藏商品信息
    - 获取收藏店铺信息
- 用户领取/查看优惠券
    - 查看可领取优惠券
    - 领取一张优惠券
    - 查看我的优惠券（未使用/已使用/已过期）
- 用户注销
    - 用户信息全部删除
- 用户搜索
    - 用户关键词搜索
- 用户查看商家列表
    - 选择排序方式
    - 选择筛选方式
- 用户查看商品列表
    - 选择排序方式
- 用户评价
    - 评价商家
- 用户下单
    - 填入地址
    - 使用优惠券
    - 进行支付

---

2. 商家模块（merchant）

- 商家注册

    - 提交店铺名、手机号、密码申请入驻
- 商家登录
    - 使用用户名+密码登录
- 商家登出
    - 退出登录
- 商家注销
    - 商家信息全部删除
- 获取当前商家信息
    - 查询店铺基础资料
- 商家修改店铺信息（在json里写出来的字段判断所要修改的字段）
    - 修改店铺名称
    - 修改店铺地址
    - 修改店铺联系方式
    - 修改店铺状态（营业中/休息中）
- 商家查看店铺评价
    - 查看所有评价
    - 筛选（用type字段区分筛选方式）
- 商家管理商品
    - 新增商品
    - 修改商品信息（名称、描述、价格、库存、分类、图像）
    - 删除商品
    - 查询商品列表（按分类、状态筛选）
- 商家设置促销活动
    - 添加促销活动（折扣、满减、套餐）
    - 修改促销活动
    - 取消促销活动
- 商家处理订单
    - 接单（确认备餐）
    - 拒单（订单取消）
    - 更新备餐完成状态
- 商家查看销售统计

- 查看店铺销售额
- 查看销量排名
- 查看商品销量分析

---

3. 骑手模块（rider）

- 骑手注册
  - 提交身份证信息+手机号+密码
- 骑手登录
  - 骑手账号密码登录
- 骑手登出
  - 退出登录
- 骑手注销
  - 删除骑手所有信息
- 获取当前骑手信息
  - 查询个人资料与接单信息
- 修改当前骑手信息
  - 修改个人资料与接单信息
- 骑手切换接单方式
  - 选择抢单或派单
- 骑手查看待抢订单
  - 列出所有附近未被接单的订单
- 骑手抢单/取消
  - 确认抢单
  - 取消订单派送
- 骑手配送中订单状态更新
  - 更新状态为 "待取餐"
  - 更新状态为 "已取餐"
  - 更新状态为 "配送中"
- 骑手完成配送
  - 完成订单并结算收入

- 骑手查看接单历史与收入
  - 查看最近接单列表
  - 查看历史总收入

---

4. 管理员模块（admin）

- 管理员登录
- 管理员登出
- 管理员查看用户列表
  - 筛选用户名/手机号/注册时间
- 管理员查看商家列表
  - 查看待审核/已通过/被拒绝商家
- 管理员查看骑手列表
  - 查看所有注册骑手
- 管理员下架违规商品/店铺（可能后期审核方式要更改：上架前审核/上架后审核）
  - 下架特定商品
  - 下架特定店铺
- 管理员查看整体平台数据统计
  - 查看用户增长趋势
  - 查看商家数量变化
  - 查看平台总交易额变化
  - 查看骑手接单量变化
- 管理员搜索
  - 关键词检索

---

5. 购物车模块（cart）

- 添加商品到购物车
- 查看购物车内商品列表
- 修改购物车商品
- 删除购物车商品或清空购物车（可多选或全选）

---

6. 消息模块（message）

- 系统通知
- 用户私聊（发送、接收消息）

三、接口文档

1. 文档概览

2. 公共约定

- 参数，方法名统一用首字母小写的驼峰
- 类名统一用驼峰命名
- 包名全部小写
- 设置工具类Response：一个泛型响应类，用于封装响应数据，包裹data和相应的成功状态和错误信息:

代码块

```
package org.demo.baoleme.dto.response;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

/**
 * 通用响应包装类，用于封装接口返回值
 * @param <T> 响应数据的类型
 */
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Response<T> {

    /** 响应数据本体 */
    private T data;

    /** 是否成功 */
    private boolean success;

    /** 错误信息（仅在失败时使用） */
```

```
23      private String errorMsg;
24
25      /**
26       * 构造一个成功响应
27       * @param data 响应数据
28       * @param <K> 数据类型
29       * @return Response<K>
30       */
31      public static <K> Response<K> newSuccess(K data) {
32          return new Response<>(data, true, null);
33      }
34
35      /**
36       * 构造一个失败响应（不包含数据）
37       * @param errorMsg 错误提示信息
38       * @param <K> 数据类型
39       * @return Response<K>
40       */
41      public static <K> Response<K> newError(String errorMsg) {
42          return new Response<>(null, false, errorMsg);
43      }
44
45      /**
46       * 构造一个失败响应（包含数据）
47       * @param data 错误时附带的数据
48       * @param errorMsg 错误信息
49       * @param <K> 数据类型
50       * @return Response<K>
51       */
52      public static <K> Response<K> newError(K data, String errorMsg) {
53          return new Response<>(data, false, errorMsg);
54      }
55  }
```

- 方法public static <K> Response<K> newSuccess(K data):创建一个新的成功的响应实例，并包含提供的数据

  @param data 要包含在成功响应中的数据

  @param <K> 数据的类型

  @return 一个新的响应实例，指示成功

- 方法public static <K> Response<K> newError(K data):*创建一个新的错误响应实例，并包含提供的数据*

  @param data 要包含在错误响应中的数据

@param <K> 数据的类型

@return 一个新的响应实例，指示失败

于是下面的返回参数应如下：

```
代码块
1   {
2     "data": {...}
3     "success":true/false
4     "errorMsg":...
5   }
```

3. 模块接口列表

3.1 用户模块接口文档

3.1.1. 用户注册接口

- **接口名称**：用户注册
- **请求地址**：`/user/register`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
username	string	是	用户名
password	string	是	密码
phone	string	是	手机号

- **返回示例**：

成功：

```
代码块
1   {
2     "success": true,
3     "message": "注册成功"
```

```
}
```

失败：

```
代码块
{
  "success": false,
  "message": "用户名已存在"
}
```

3.1.2. 用户登录接口

- **接口名称**：用户登录
- **请求地址**：`/user/login`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
phone	string	是	手机号
password	string	是	密码

- **返回示例**：

成功：

```
代码块
{
  "success": true,
  "token": "jwt-token",
  "role": "user"
}
```

失败：

```
代码块
```

```
{
  "success": false,
  "message": "用户名或密码错误"
}
```

3.1.3. 用户登出接口

- **接口名称**：用户登出
- **请求地址**：`/user/logout`
- **请求方式**：POST
- **请求参数**：无（需在header中带token）
- **返回示例**：

代码块

```
{
  "success": true,
  "message": "退出成功"
}
```

3.1.4. 获取当前用户信息接口

- **接口名称**：获取信息
- **请求地址**：`/user/info`
- **请求方式**：GET
- **请求参数**：无（需在header中带token）
- **返回示例**：

```
{
  "success": true,
  "data": {
    "id": 1,
    "username": "testuser",
    "gender":"男",
    "phone": "12345678901",
    "avatar": "https://example.com/avatar.png"
  }
}
```

3.1.5. 修改用户信息接口

- **接口名称**：修改信息
- **请求地址**：`/user/update`
- **请求方式**：POST
- **请求参数**（可以只传需要修改的字段）：

参数名	类型	是否必填	说明
username	string	否	修改后的用户名
phone	string	否	修改后的手机号
password	string	否	修改后的密码

- **返回示例**：

代码块

```
{
  "success": true,
  "message": "修改成功"
}
```

3.1.6. 用户订单记录接口

- **接口名称**：查看订单记录
- **请求地址**：`/user/history`
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "product_id": 1,
      "product_name": "奶茶",
      "create_time": "2025-04-01T12:00:00"
    },
    {
      "store_id": 2,
      "store_name": "汉堡店",
      "create_time": "2025-04-01T13:00:00"
    }
  ]
}
```

3.1.7. 用户收藏店铺接口

- **接口名称**：收藏店铺
- **请求地址**：`/user/favorite/store`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
store_id	bigint	是	店铺ID

- **返回示例：**

```
代码块
1  {
2    "success": true,
3    "message": "收藏成功"
4  }
```

3.1.8. 用户浏览收藏店铺接口

- **接口名称：**浏览收藏店铺
- **请求地址：**`/user/favorite/watc/store`
- **请求方式：**GET
- **请求参数：**
- **返回示例：**

```
代码块
1  {
2    "success": true,
3    "data": [
4        {
5            "store_id": 1,
6            "store_name": "饱了么商家"
7        }
8    ]
9  }
```

3.1.9. 用户查看优惠券接口

- **请求地址**：`/user/coupons`
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "coupon_id": 1,
      "code": "SAVE10",
      "discount": 10,
      "expiration_date": "2025-05-01"
    }
  ]
}
```

3.1.10.用户领取优惠券

- **请求地址**：`/user/coupon/claim`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
coupon_id	bigint	是	优惠券ID

- **返回示例**：

代码块

```
{
  "success": true,
  "message": "领取成功"
}
```

3.1.11. 用户注销接口（可以先不实现）

- **接口名称**：用户注销
- **请求地址**：`/user/cancel`
- **请求方式**：POST
- **请求参数**：无（需在header中带token）
- **返回示例**：

代码块

```
{
  "success": true,
  "message": "注销成功"
}
```

3.1.12. 用户查看当前订单接口

- **接口名称**：查看订单记录
- **请求地址**：`/user/current`
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "product_id": 1,
      "product_name": "奶茶",
      "create_time": "2025-04-01T12:00:00"
    },
    {
      "store_id": 2,
      "store_name": "汉堡店",
```

```
        "create_time": "2025-04-01T13:00:00"
      }
    ]
    "predict_time":"2025-04-01T13:30:00"
  }
```

3.1.13 用户搜索接口

- **接口名称**：全局搜索
- **请求地址**：`/user/search`
- **请求方式**：GET
- **请求参数**：

参数名	类型	是否必填	说明
keyword	string	是	搜索关键词（商品名/店铺名）
page	int	否	分页页码（默认1）
size	int	否	每页数量（默认10）

- **返回示例**：

代码块

```
{
  "success": true,
  "data": {
    "products": [
      {
        "product_id": 1,
        "product_name": "奶茶",
        "price": 12.5,
        "shop_name": "茶颜悦色"
      }
    ],
    "shops": [
      {
        "shop_id": 1,
        "shop_name": "汉堡店",
        "rating": 4.8
      }
```

```
    ],
    "total": 15
  }
}
```

3.1.14 用户查看商家列表接口

- **接口名称**：商家列表
- **请求地址**：`/user/shops`
- **请求方式**：GET
- **请求参数**：

参数名	类型	是否必填	说明
type	string	否	店铺类型（如快餐、饮品）
min_rating	float	否	最低评分（默认0）
max_price	float	否	最高人均价格
page	int	否	分页页码（默认1）
size	int	否	每页数量（默认10）

- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "shop_id": 1,
      "shop_name": "汉堡店",
      "type": "快餐",
      "rating": 4.8,
      "delivery_time": "30分钟",
      "image": "https://example.com/shop1.png"
    }
```

```
    ],
    "total": 20
  }
```

3.1.15 用户查看商品列表接口

- **接口名称**：商品列表
- **请求地址**：`/user/products`
- **请求方式**：GET
- **请求参数**：

参数名	类型	是否必填	说明
shop_id	bigint	是	店铺ID
category	string	否	商品分类（如饮品、主食）
min_price	float	否	最低价格
max_price	float	否	最高价格
sort_by	string	否	排序方式（sales/price）

- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "product_id": 1,
      "product_name": "珍珠奶茶",
      "price": 15.0,
      "sales": 200,
      "image": "https://example.com/product1.png"
    }
  ]
}
```

3.1.16 用户评价接口

- **接口名称**：提交评价
- **请求地址**：`/user/review`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
order_id	bigint	是	订单ID
rating	int	是	评分（1-5星）
comment	string	否	评论内容
images	array	否	图片URL列表（最多3张）

- **返回示例**：

```
代码块
1  {
2    "success": true,
3    "message": "评价提交成功"
4  }
```

3.1.17 用户下单接口

- **接口名称**：提交订单
- **请求地址**：`/user/order`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
address_id	bigint	是	收货地址ID
coupon_id	bigint	否	优惠券ID
remark	string	否	订单备注

- **返回示例**：

```
代码块
1  {
2    "success": true,
3    "data": {
4      "order_id": "20250401123456",
5      "total_price": 68.5,
6      "status": "待支付",
7      "pay_url": "https://pay.example.com/order=xxx"
8    }
9  }
```

3.2 商家模块接口文档

3.2.1. 商家注册

- **接口名称**：商家注册
- **请求地址**：`/merchant/register`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
username	string	是	用户名
password	string	是	密码
phone	string	否	手机号

- 请求体：

```
代码块
1  {
2    "username": "天天生鲜",
3    "password": "Merchant123",
4    "phone": "13812345678"
```

```
}
```

- **返回示例**：

成功：

```
代码块
{
  "code": 200,
  "data": {
    "id": "。。。",
    "created_at": "2023-10-01 14:30:00"
  },
  "message": "注册成功"
}
```

失败：

```
代码块
{
  "code": 400,
  "data": null,
  "message": "手机号已被注册"
}
```

3.2.2. 商家登录

- **接口名称**：商家登录
- **请求地址**：`/merchant/login`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
username	string	否（用户名手机号二选一）	用户名
phone	string	否	手机号
password	string	是	密码

- 请求体：

```
代码块
1  {
2    "username": "天天生鲜",
3    "phone": "111 1111 1111",
4    "password": "Merchant123"
5  }
```

- **返回示例**：

成功：

```
代码块
1  {
2    "code": 200,
3    "data": {
4        "id": "20000001",
5        "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
6        "expires_in": 7200    // token有效期（秒）
7    },
8    "message": "登录成功"
9  }
```

失败：

```
代码块
1  {
2      "code": 401,
3      "data": null,
4      "message": "手机号或密码错误"
5  }
```

3.2.3. 商家登出

- **接口名称**：商家登出
- **请求地址**：`/merchant/logout`
- **请求方式**：POST
- **请求参数**：无（需在header中带token）

- **返回示例**：

```
代码块
1  {
2    "code": 200,
3    "data": {},
4    "message": "登出成功"
5  }
```

3.2.4. 商家注销

- **接口名称**：商家注销
- **请求地址**：`/merchant/deactivate`
- **请求方式**：POST
- **请求参数**：无（需在header中带token）
- **返回示例**：

```
代码块
1  {
2    "code": 200,
3    "data": {},
4    "message": "账号已注销"
5  }
```

3.2.5. 获取当前商家信息

- **接口名称**：获取信息
- **请求地址**：`/merchant/info`
- **请求方式**：GET
- **请求参数**：无（需在header中带token）
- **返回示例**：

```
代码块
1  {
2    "code": 200,
3    "data": {
4      "id": "M20231001123456",
5      "username": "天天生鲜",
6      "phone": "138****5678",
```

```
    "created_at": "2023-10-01T14:30:00+08:00"
  },
  "message": "success"
}
```

3.2.6. 商家修改自身信息

- **接口名称**：修改商家信息
- **请求地址**：`/merchant/`info
- **请求方式**：PATCH
- **请求参数（需在header中带token）**（可以只传需要修改的字段）：

参数名	类型	是否必填	说明
username	string	否	修改后的用户名
phone	string	否	修改后的手机号
password	string	否	修改后的密码
avatar	string	否	头像

- 请求体：

代码块

```
{
  "name": "张三"
}
```

- **返回示例**：

代码块

```
{
  "code": 200,
  "data": {
    "id": 20231001123456,
    "name": "张三",
    "phone": "111 1111 1111"
  },
  "message": "信息更新成功"
}
```

3.2.7.商家管理店铺

3.2.7.1 修改店铺信息

- **接口名称**：修改店铺信息
- **请求地址**：`/shop/info`
- **请求方式**：PATCH
- **请求参数（需在header中带token）**（可以只传需要修改的字段）：
- 请求体：

```
代码块
1  {
2    "name": "天天生鲜旗舰店",
3    "location": "北京市海淀区中关村大街1号"
4  }
```

- **返回示例**：

```
代码块
1  {
2    "code": 200,
3    "data": {
4      "id": 20231001123456,
5      "name": "天天生鲜旗舰店",
6      "desc": "生鲜零售",
7      "location": "北京市海淀区中关村大街1号",
8      "rating": 4.8,
9      "status": 1,
10     "created_at": "2023-10-01T14:30:00+08:00"
11   },
12   "message": "店铺信息更新成功"
13 }
```

3.2.7.2商家创建店铺

3.2.7.3商家删除店铺

3.2.7.4查看店铺信息

3.2.8. 商家查看店铺评价

- **接口名称**：商家查看店铺评价
- **请求地址**：/review/list
- **请求方式**：GET
- **请求参数**（可以只传需要修改的字段）：

参数名	类型	是否必填	说明
store_id	bigint	是	有效店铺ID
type	int	否	筛选类型（好评，差评，带图）
page	int	否	分页页码（默认 1）
page_size	int	否	每页数量（默认 10）

- **返回示例**：

代码块

```
{
  "code": 200,
  "data": {
    "total": 15,
    "comments": [
      {
        "user_avatar": "https://example.com/avatar.jpg",
        "username": "用户1",
        "content": "味道很好",
        "rating": 5,
        "images": ["url1", "url2"],
        "create_time": "2025-04-20T10:00:00"
      }
    ]
  }
  "message": "success"
}
```

3.2.9. 商家管理商品

3.2.9.1 新增商品

- **接口名称**：新增商品
- **请求地址**：`/products`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
name	string	是	商品名称
descripition	string	是	商品描述
price	double	是	商品价格
stock	integer	是	商品库存
category	string	是	商品标签
image	string	否	图片

- **返回示例**：

代码块

```
{
  "code": 200,
  "data": {
    "product_id": 123
  }
  "message": "success"
}
```

3.2.9.2 修改商品信息

- **接口名称**：修改商品
- **请求地址**：`/products/updateInfo/`
- **请求方式**：PUT
- **请求参数**：

参数名	类型	是否必填	说明
name	string	是	商品名称

descripition	string	是	商品描述
price	double	否	商品价格
stock	integer	是	商品库存
category	string	是	商品标签
status	Product.ProductStatus	否	上架/下架
image	string	否	图片

- **返回示例：**

代码块

```
{
  "code": 200,
  "data": {
    商品信息
  },
  "message": "修改成功"
}
```

3.2.9.3 查看商品列表

- **接口名称**：查看商品
- **请求地址**：/product/list
- **请求方式**：GET
- **请求参数**：token

参数名	类型	是否必填	说明
shore_id	long	是	
status	bool	否	商品状态（上架/下架）
category_id	Product.ProductStatus	否	分类ID
page	integer		
pageSize	integer		

- **返回示例：**

代码块

```
{
  "code": 200,
  "data": [
    {
      "id": 123,
      "name": "奶茶",
      "price": 15.0,
      "month_sales": 102,
      "rating": 4.8
    }
  ],
  "message": "success"
}
```

3.2.9.4.商家删除商品

3.2.9.5.查看商品信息

3.2.10. 商家设置促销活动

3.2.10.1 创建促销活动

- **接口名称：**创建活动
- **请求地址：** `/promotions/create/`
- **请求方式：**POST
- **请求参数：**

代码块

```
{
  "type": "DISCOUNT",  // DISCOUNT-折扣 FULL_REDUCE-满减
  "desc": "五一特惠",
  "discount": 0.9,     // 折扣时必填
  "full_amount": 100,  // 满减时必填
  "reduce_amount": 20,
  "start_at": "2025-05-01T00:00:00",
  "end_at": "2025-05-07T23:59:59"
}
```

- **返回示例：**

```
{
    "code": 200,
    "data": {
      "promotion_id": 456
    }
    "message": "success"
}
```

3.2.10.2 活动列表

- **接口名称：**活动列表
- **请求地址：**`/promotions/list/`
- **请求方式：**GET
- **请求参数：**token, shop_id
- **返回示例：**

代码块

```
{
    "code": 200,
    "data": [
      {
        "id": 456,
        "name": "五一特惠",
        "status": "RUNNING",  // NOT_STARTED/RUNNING/ENDED
        "discount": "9折",
        "time_range": "5.1-5.7"
      }
    ]
    "message": "success"
}
```

3.2.10.3 删除活动

3.2.10.4 修改活动

3.2.11. 商家处理订单

3.2.11.1 订单列表

- **接口名称：**订单列表
- **请求地址：**`/orders/get/`

- **请求方式：**GET
- **请求参数：**

参数名	类型	是否必填	说明
shop_id			
status	string	否	订单状态（WAITING-待接单 PROCESSING-制作中）

- 返回示例：

代码块

```
{
  "code": 200,
  "data": [
    {
      "order_id": "ORDER20250410001",
      "user_name": "用户A",
      "total_amount": 38.5,
      "status": "WAITING",
      "create_time": "2025-04-10T12:30:00"
    }
  ]
  "message": "success"
}
```

3.2.10.2 订单操作

- **接口名称：**订单操作
- **请求地址：/order/updateStatus**
- **请求方式：PATCH**
- **请求参数：header中带有token**

代码块

```
{
  "order_id": 20231001123456,
  "new_status": 4,    // 只能为 2配送中 或 4取消
  "cancel_reason": "食材供应不足"
}
```

- **返回示例：**

```
代码块
1  {
2    "code": 200,
3    "data": {
4      "order_id": "OD202310011234",
5      "old_status": 1,
6      "new_status": 2,
7      "updated_at": "2023-10-05T15:30:00+08:00"
8    },
9    "message": "状态更新成功"
10 }
```

3.2.12. 商家查看销售统计

3.2.12.1 销售概况

- **接口名称：**销售概况
- **请求地址：/sales/stats/overview**
- **请求方式：**GET
- **请求参数：**

参数名	类型	是否必填	说明
store_id	long	是	
time_range	TimeRange	否	时间范围（TODAY/WEEK/MONTH）

- 返回示例：

```
代码块
1  {
2    "code": 200,
3    "data": {
4      "total_sales": 12580.5,
5      "order_count": 156,
6      "popular_products": [
7        {
```

```
        "product_name": "珍珠奶茶",
        "sales_count": 89
      }
    ]
  },
  "message": "success"
}
```

3.2.12.2 销售趋势

- **接口名称：**销售趋势
- **请求地址：/sales/stats/trend**
- **请求方式：**GET
- **请求参数：**

参数名	类型	是否必填	说明
type	TimeLine	是	DAY-按日 WEEK-按周 MONTH-按月
num_of_recent_days	int	否	最近天数（默认7）

- 返回示例：

代码块

```
{
  "code": 200,
  "data": {
    "dates": ["04-01", "04-02"],
    "values": [1250, 1800]
  }
  "message": "success"
}
```

3.3 骑手模块接口文档

以下是基于功能需求的骑手接口设计，采用 RESTful 风格实现：

3.3.1 骑手注册

- **接口名称**：骑手注册
- **请求地址**：/rider/register
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
`username`	string	true	用户名
`phone`	string	true	手机号
`password`	string	true	密码

- **返回示例**：

代码块

```
{
    "success": true,
    "message": "success",
    "data": {
        "user_id": 30000005,
        "username": "Mxi_test",
        "phone": "12222222222"
    },
    "code": 200
}
```

3.3.2 骑手登录

- **接口名称**：骑手登录
- **请求地址**：/rider/login
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
`phone`	string	true	手机号
`password`	string	true	密码

- **返回示例**：

代码块

```
{
    "success": true,
    "message": "success",
    "data": {
        "token":
"eyJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoicmlkZXIiLCJ1c2VyX2lkIjozMDAwMDAwNCwidXNlcm5hbWUiOiJNeGkiLCJpYXQiOjE3NDc0NTIyOTYsImV4cCI6MTc0NzUzODY5Nn0.c2dYybxvjSPEV_ZHK3hVmHAjP1iVbGoG_-a4C4SjWio",
        "username": "Mxi",
        "user_id": 30000004
    },
    "code": 200
}
```

3.3.3 骑手登出

- **接口名称**：骑手登出
- **请求地址**：/rider/logout
- **请求方式**：POST
- **请求参数**：无
- **返回示例**：

代码块

```
{
    "success": true,
    "message": "success",
    "code": 200
}
```

3.3.4 骑手注销

- **接口名称**：骑手注销
- **请求地址**：/rider/delete
- **请求方式**：DELETE
- **请求参数**：无
- **返回示例**：

```
代码块
    "success": true,
    "message": "success",
    "code": 200
}
```

3.3.5 获取当前骑手信息

- **接口名称**：获取当前骑手信息
- **请求地址**：/rider/info
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

```
代码块
{
    "success": true,
    "message": "success",
    "data": {
        "user_id": 30000004,
        "username": "Mxi",
        "phone": "11111111111",
        "order_status": 1,
        "dispatch_mode": 1,
        "balance": 0
    },
    "code": 200
}
```

3.3.6 修改当前骑手信息

- **接口名称**：修改当前骑手信息
- **请求地址**：/rider/update
- **请求方式**：PUT
- **请求参数**：

参数名	类型	是否必填	说明
`username`	string	false	新用户名
`password`	string	false	新密码

`phone`	string	false	新手机号
`order_status`	Integer	false	新的接单状态
`dispatch_mode`	Integer	false	新接单方式

- **返回示例：**

代码块

```
{
    "success": true,
    "message": "success",
    "code": 200
}
```

3.3.7 查询可抢订单

- **请求地址**：/orders/available
- **请求方式**：GET
- **请求头**：
  - Authorization: Bearer <骑手Token>
- **请求参数**：无
- **返回示例：**

代码块

```
{
    "success": true,
    "message": "success",
    "data": {
        "orders": [
            {
                "id": 70000002,
                "user_id": 10000002,
                "store_id": 40000002,
                "rider_id": null,
                "status": 0,
                "total_price": 25.00,
                "created_at": "2025-05-16T19:57:46",
                "deadline": "2025-05-16T20:42:46",
                "ended_at": null
            }
        ]
```

```
    },
    "code": 200
}
```

3.3.8 抢单

- **请求地址**：/orders/grab
- **请求方式**：PUT
- **请求头**：
  - Authorization: Bearer <骑手Token>
- **请求参数**：

参数名	类型	是否必填	说明
order_id	Long	是	订单ID

- **请求体示例**：

代码块

```
{
  "order_id": 70000002
}
```

- 返回体示例

代码块

```
{
    "success": true,
    "message": "success",
    "data": {
        "order_id": 70000002,
        "status": "ACCEPTED",
        "pickup_deadline": "2025-05-17T17:32:11.668752"
    },
    "code": 200
}
```

3.3.9 取消订单配送

- **请求地址**：/orders/cancel
- **请求方式**：PUT
- **请求头**：
  - Authorization: Bearer <骑手Token>
- **请求参数**：

参数名	类型	是否必填	说明
order_id	Long	是	订单ID

- **请求体示例**：

```
代码块
1  {
2    "order_id": 70000001
3  }
```

- 返回体示例

```
代码块
1  {
2      "success": true,
3      "message": "success",
4      "data": {
5          "order_id": 70000002,
6          "status": "CANCELLED"
7      },
8      "code": 200
9  }
```

3.3.10 更新订单状态

- **请求地址**：/orders/rider-update-status
- **请求方式**：POST
- **请求头**：
  - Authorization: Bearer <骑手Token>

- **请求参数**：

参数名	类型	是否必填	说明
order_id	Long	是	订单ID
target_status	Integer	是	目标状态（如 2 配送中）

- **请求体示例**：

代码块

```
{
  "order_id": 70000002,
  "target_status": 3,
}
```

- 返回体示例

代码块

```
{
    "success": true,
    "message": "success",
    "data": {
        "order_id": 70000002,
        "status": 3,
        "updated_at": "2025-05-17T18:50:15.191527"
    },
    "code": 200
}
```

3.3.11 查询骑手订单记历史记录

- **请求地址**：/orders/rider-history-query
- **请求方式**：POST
- **请求头**：
  - Authorization: Bearer <骑手Token>
- **请求参数**：

参数名	类型	是否必填	说明
status	Integer	否	订单状态筛选
start_time	String	否	起始时间
end_time	String	否	结束时间
page	int	是	页码
page_size	int	是	每页条数

- **请求体示例**：

代码块

```
{
  "status": 3,
  "start_time": "2023-08-01T00:00:00",
  "end_time": "2023-08-20T23:59:59",
  "page": 1,
  "page_size": 10
}
```

- 返回体示例

代码块

```
{
    "success": true,
    "message": "success",
    "data": {
        "orders": [
            {
                "order_id": 70000002,
                "status": 3,
                "total_amount": 25.00,
                "completed_at": "2025-05-17T18:50:15"
            }
        ]
    },
    "code": 200
}
```

3.3.12 骑手收入统计

- **请求地址**：/orders/rider-history-query
- **请求方式**：GET

- **请求头**：
- Authorization: Bearer <骑手Token>
- **请求参数**：无
- 返回体示例

代码块

```
{
    "success": true,
    "message": "success",
    "data": {
        "total_earnings": 25.00,
        "current_month": 25.00,
        "completed_orders": 1
    },
    "code": 200
}
```

技术说明

1. **身份验证**：所有接口需通过 JWT Token 验证
2. **状态管理**：使用有限状态机控制订单流转
3. **地理位置**：使用 Redis GEO 实现附近订单查询
4. **并发控制**：使用乐观锁处理抢单竞争

错误码示例：

- 400：参数校验失败
- 401：未授权访问
- 429：频繁操作限制

接口设计可根据具体技术栈调整实现细节，建议配合 WebSocket 实现订单状态实时推送。

3.4 管理员模块接口文档

3.4.1. 管理员登录接口

- **接口名称**：管理员登录
- **请求地址**：`/admin/login`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
admin_id	long	是	id
password	string	是	密码

- **返回示例**：

成功：

```
代码块
1  {
2    "success": true,
3    "token": "jwt-token",
4    "role": "admin"
5  }
```

失败：

```
代码块
1  {
2    "success": false,
3    "token": "jwt-token",
4    "message": "用户名或密码错误"
5  }
```

3.4.2. 管理员登出接口

- **接口名称**：管理员登出
- **请求地址**：`/admin/logout`
- **请求方式**：POST
- **请求参数**：无（需在header中带token）
- **返回示例**：

代码块

```
{
  "success": true,
  "token": "jwt-token",
  "message": "退出成功"
}
```

3.4.3. 管理员查看用户列表

- **接口名称**：管理员查看用户列表
- **请求地址**：`/admin/userlist`
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
      {
          "id": 1,
          "username":baoleme
      }
  ]
  "message": "退出成功"
```

```
10  }
```

3.4.4. 管理员查看店铺列表

- **接口名称**：管理员查看店铺列表
- **请求地址**：`/admin/shoplist`
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
1   {
2     "success": true,
3     "data": [
4         {
5             "id": 1,
6             "username":baoleme
7         }
8     ]
9     "message": "退出成功"
10  }
```

3.4.5. 管理员查看骑手列表

- **接口名称**：管理员查看骑手列表
- **请求地址**：`/admin/riderlist`
- **请求方式**：GET
- **请求参数**：无

- **返回示例**：

```
代码块
1  {
2    "success": true,
3    "data": [
4        {
5            "id": 1,
6            "username":baoleme
7        }
8    ]
9    "message": "退出成功"
10 }
```

3.4.6. 管理员下架店铺/商品接口

- **接口名称**：管理员下架店铺/商品
- **请求地址**：`/admin/delete`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
shopname	string	是	店铺名
foodname	string	否	菜品名

- **返回示例**：

成功：

```
代码块
1  {
2    "success": true,
3    "token": "jwt-token",
4    "role": "admin"
```

```
5   }
```

3.4.7. 管理员查看平台信息

- **接口名称**：管理员查看平台信息
- **请求地址**：`/admin/check`
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
      {
          "userNumber": 1,
          "shopNumber": 1,
          "totalAmount": 1,
          "totalOrder": 1
      }
  ]
  "message": "退出成功"
}
```

3.4.8. 管理员搜索接口

- **接口名称**：管理员搜索
- **请求地址**：`/admin/find`
- **请求方式**：POST

- **请求参数**：

参数名	类型	是否必填	说明
keyword	string	是	关键词

- **返回示例**：

```
代码块
1  {
2    "success": true,
3    "data": [
4        {
5            "shopName": "baleme",
6            "shopId": 1,
7            "foodNmae": "food",
8            "foodId": 1
9        }
10       {
11           ...
12       }
13   ]
14   "role": "admin"
15 }
```

3.5 购物车模块接口文档

3.5.1 添加商品到购物车

- **接口名称**：添加商品到购物车
- **请求地址**：/cart/add
- **请求方式**：POST
- **请求参数**：
  - `product_id`：string, 商品ID
  - `quantity`：integer, 商品数量
- **返回示例**：

```
代码块
1  {
2    "success": true,
```

```
  "errorMsg":
}
```

3.5.2 查看购物车内商品列表

- **接口名称**：查看购物车内商品列表
- **请求地址**：/cart/view
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "product_id": 101,
      "product_name": "商品A",
      "quantity": 2,
      "price": 20.00
    },
    {
      "product_id": 102,
      "product_name": "商品B",
      "quantity": 1,
      "price": 50.00
    }
  ],
  "errorMsg":
}

```

3.5.3 修改购物车商品

- **接口名称**：修改购物车商品
- **请求地址**：/cart/update
- **请求方式**：PUT
- **请求参数**：
  - `product_id`：string, 商品ID
  - `quantity`：integer, 新商品数量

- **返回示例**：

```
代码块
1  {
2    "success": true,
3    "errorMsg":
4  }
```

3.5.4 删除购物车商品或清空购物车

- **接口名称**：删除购物车商品或清空购物车
- **请求地址**：/cart/remove
- **请求方式**：DELETE
- **请求参数**：
  - `product_ids`：array, 商品ID数组（若删除单个商品，传递该商品ID；若清空购物车，传递空数组）
- **返回示例**：

```
代码块
1  {
2    "success": true,
3    "errorMsg":
4  }
```

3.6 消息模块接口文档

3.6.1 系统通知

3.6.1.1 获取系统通知列表

- **接口名称**：获取系统通知
- **请求地址**：`/messages/notifications`
- **请求方式**：GET
- **请求参数**：

参数名	类型	是否必填	说明

is_read	boolean	否	是否已读（true-已读 false-未读）
page	int	否	分页页码（默认1）
page_size	int	否	每页数量（默认10）

- **返回示例**：

```
代码块
1  {
2    "success": true,
3    "data": {
4      "unread_count": 5,
5      "list": [
6        {
7          "id": "NT20240410001",
8          "title": "系统维护通知",
9          "content": "平台将于今晚00:00-02:00进行系统升级",
10         "is_read": false,
11         "create_time": "2025-04-10T15:30:00",
12         "type": "SYSTEM" // SYSTEM-系统通知 ORDER-订单通知 COUPON-优惠券通知
13       }
14     ]
15   }
16 }
```

3.6.1.2 标记通知为已读

- **接口名称**：标记已读
- **请求地址**：`/messages/notifications/read`
- **请求方式**：POST
- **请求参数**：

参数名	类型	是否必填	说明
message_ids （NT+date+number）	string	是	["NT20240410001", "NT20240410002"] NT：通知类型

- **返回示例**：

```
代码块
2     "success": true,
3     "message": "标记成功"
4   }
```

3.6.2 用户私聊（实时好不好实现？能不能存储到数据库然后每隔一段时间刷新来接收）

3.6.2.1 发送私信

- **接口名称**：发送消息
- **请求地址**：/messages/private
- **请求方式**：POST
- **请求参数**：

```
代码块
1   {
2     "receiver_id": "user123",
3     "content": "您好，请问商品什么时候发货？",
4     "type": "TEXT" // TEXT-文本 IMAGE-图片（预上传才行
5   }
```

- **返回示例**：

```
代码块
1   {
2     "success": true,
3     "data": {
4       "message_id": "MSG20240410001",
5       "send_time": "2025-04-10T16:20:00"
6     }
7   }
```

3.6.2.2 获取聊天记录

- **接口名称**：获取聊天记录
- **请求地址**：/messages/private/history
- **请求方式**：GET
- **请求参数**：

参数名	类型	是否必填	说明

target_user_id	string	是	对方用户ID
before_time	string	否	查询此时间之前的记录（ISO格式）
limit	int	否	返回条数（默认20，最大100）

- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "id": "MSG20240409001",
      "sender_id": "user123",
      "content": "明天能送到吗？",
      "type": "TEXT",
      "create_time": "2025-04-09T10:15:00",
      "is_self": false // 是否当前用户发送
    }
  ]
}
```

3.6.2.3 获取会话列表

- **接口名称**：会话列表
- **请求地址**：/messages/conversations
- **请求方式**：GET
- **请求参数**：无
- **返回示例**：

代码块

```
{
  "success": true,
  "data": [
    {
      "user_id": "user123",
      "username": "张先生",
      "avatar": "https://example.com/avatar.jpg",
      "last_message": {
```

```
          "content": "好的，我会尽快发货",
          "time": "2025-04-10T11:30:00",
          "unread_count": 3
        }
      }
    ]
  }
```