

《设计模式实践》大作业文档

题 目： 在线商品销售平台（商家端）

作 者： 徐少奇、倪宇志、白晨、陈思恩

学 号： 2019229050、2019229051、2019559052、2019229053

学 院： 国际工程师学院

专 业： 计算机技术

指导教师： 李罡

2020 年 6 月

目 录

1 引言	1
1.1 编写目的	1
1.2 背景	1
1.3 定义	2
1.4 参考资料	2
2 需求分析	3
2.1 功能需求	3
2.2 性能需求	4
2.3 可靠性、可用性需求及出错处理需求	5
2.4 接口需求和约束	5
2.5 将来可能出现的需求	6
3 数据库设计	7
3.1 数据表结构	7
3.2 数据库设计	10
4 系统设计与实现	17
4.1 系统架构	17
4.1.1 简介	17
4.1.2 系统功能	18
4.1.3 项目技术	18
4.1.4 开发方案	21
4.1.5 部署方案	23
4.1.6 上线方案	23
4.2 基础系统	26
4.2.1 mall-db	26
4.2.2 mall-core	30
4.3 管理后台	32
4.3.1 mall-admin-api	33
4.3.2 mall-admin	36
4.4 前后端接口设计	38
4.4.1 请求格式	38
4.4.2 响应格式	39
4.4.3 错误码	41
4.4.4 Token	42

4.4.5 API 服务列表	44
5 系统使用与测试.....	46
5.1 用户管理模块测试.....	46
5.2 商场&商品管理模块测试.....	48
5.3 系统管理模块测试.....	53
5.4 配置管理模块测试.....	54
5.5 统计报表模块测试.....	55

1 引言

近几年，随着我国经济实力的不断增强，以及互联网技术的稳步发展，网购逐渐成为人们口中的热门话题。在信息时代的强力推动下，在线购物正在以惊人的速度发展。在经济全球化背景下，互联网的全球性，使得一种与地域、空间无关的世界一体化市场正在形成。基于这样的时代背景，一种全新的、基于计算机网络的新型商业机制正在逐步形成——电子商务。因此，对于 Web 开发者来说，建设一个功能完善、可用性强、能够帮助商家及时了解用户需求，以及帮助商家统计相关 B2C 订单信息的在线商品销售平台是适应目前潮流的一个热门开发方向；而对于商业管理者来说，合理有效地运营在线商品销售平台或网站也是目前主流的商业方向。

本文档为本次《设计模式实践》大作业的在线商品销售平台（商家端）而编写，详细地阐述了在线商品销售平台（商家端）的定位和设计方案，对系统中的各项功能模块、技术需求、实现环境以及所使用的实现技术进行了明确定义。

1.1 编写目的

本文档的编写目的主要有三点，第一点是阐述我们对于该平台的设计思路，包括需求分析、接口与数据库、系统架构等方面；第二点则是讲解我们将该平台进行实现的过程，包括基础系统、管理后台等方面的内容；第三点则是提供平台的使用和测试说明，方便使用者后续进行部署等操作。

本文档按照设计顺序，先进行需求分析，确定我们需要满足哪些需求，从而对功能需求进行确认；在了解到需要实现那些需求之后，开始对系统的调用接口、数据库、系统架构等规则进行设计，从而确保开发过程有序进行；随后依照设计的规则对系统进行实现，并在文档中阐述相关实现过程；最后对系统实现的功能进行测试，并在文档中对这些相关功能进行介绍。通过这样几个步骤，既可以保证系统的可靠性，同时也能使代码可读性强、容易理解、容易测试、容易修改和维护，充分而有效地将本学期所学《设计模式》课程的相应知识融入工程，转化为成果。

1.2 背景

待开发的软件系统名称：在线商品销售平台（商家端）

本项目的任务提出者：李罡 老师

本项目的任务开发者：徐少奇、倪宇志、白晨、陈思恩

本项目的目标用户：中小型电子商务运营者

1.3 定义

本文档中涉及的专业术语定义或解释，以下表形式给出，如表 1-1 所示。

表 1-1 术语定义或解释表

序号	术语名称	术语定义
1	B2C	Business-to-Consumer，商对客 电子商务模式
2	API	Application Programming Interface，应用程序接口
3	SQL	Structured Query Language，结 构化查询语言
4	JSON	JavaScript Object Notation，JS 对象简谱

1.4 参考资料

- [1] 结城浩, 杨文轩. 图解设计模式[M]. 人民邮电出版社, 2015.
- [2] AlanShalloway, JamesR.Trott, 沙洛韦,等. 设计模式精解[M]. 机械工业出版社, 2006.
- [3] fErichGamma, 加马, 李英军. 设计模式:可复用面向对象软件的基础[M]. 机械工业出版社, 2000.
- [4] 周之英. 现代软件工程. 中,基本方法篇[M]// 现代软件工程. 中, 基本方法篇. 科学出版社, 2000.
- [5] 汪云飞. JavaEE 开发的颠覆者 : Spring Boot 实战[M].

2 需求分析

本文档记录的需求分析，是软件系统处于筹备计划阶段的活动。该阶段主要分析系统在功能上需要实现什么，而暂时不考虑考虑如何去实现。需求分析的目标是把用户对待开发软件提出的“要求”或“需要”进行分析与整理，确认后形成描述完整、清晰与规范的文档，确定软件需要实现哪些功能，完成哪些工作。此外，本文档的需求分析还包括一些非功能性需求，如软件性能、可靠性、响应时间、可扩展性，以及软件设计的约束条件等内容，以便后续进行设计时可以进行全面详细的考虑。

2.1 功能需求

本系统的功能需求划分为如下几点：

1.用户管理

（1）会员管理：查找指定名称或手机号的会员，显示用户的详细信息，对用户信息进行导出等；

（2）收货地址：查找指定 ID 或收货人名称的地址，显示地址的详细信息，对地址信息进行导出等；

（3）会员收藏：查找指定用户或指定商品的收藏信息，显示收藏的详细信息，对收藏信息进行导出等；

（4）会员足迹：查找指定用户或指定商品的浏览记录，显示浏览记录的详细信息，对浏览信息进行导出等；

（5）搜索历史：查找指定用户或关键字的搜索记录，显示搜索记录的详细信息，对搜索信息进行导出等；

（6）意见反馈：查找指定用户或反馈 ID 的反馈记录，显示反馈记录的详细信息，对反馈信息进行导出等；

2.商场&商品管理

（1）商场信息管理：发货行政区域的查询功能，品牌制造商和商品类目的增删改查功能；

（2）订单管理：查找指定用户或订单 ID 或订单状态的订单记录，显示订单记录的详细信息，对订单信息进行导出，打印订单信息等；

（3）通用问题：设置通用问题内容的快捷回复，查询指定关键词的通用问题，对通用问题进行增删改处理，对通用问题进行导出等；

（4）关键词：设置关键词对应的推广链接，开启或关闭推荐状态，查询指定

关键词或跳转链接对应的推广内容,对推广内容进行增删改处理,对关键词和跳转链接进行导出等;

(5) 商品列表: 查找指定编号或名称对应的商品信息,显示商品记录的名称、图片、价格、在售、规格、库存等一系列详细信息,对商品列表进行导出、更新和删除等;

(6) 商品上架: 新增商品的信息;

(7) 商品评论: 查找指定用户或商品对应的商品评论,显示商品评论的打分和评论内容等信息,对商品评论进行回复或删除操作;

3. 系统管理

(1) 管理员: 对管理员进行增删改查操作,为管理员添加或删除不同角色,导出管理员记录等;

(2) 操作日志: 显示后台管理系统的一系列操作行为日志,包括操作角色、IP 地址、时间、相关操作等均被记录在日志中,并提供操作日志的查找筛选;

(3) 角色管理: 对角色进行增删改查操作,为角色授予或取消授予不同的后台管理权限;

(4) 对象存储: 按照对象 KEY 或对象名称进行对象数据的查找,显示对象存储的详细信息,对对象存储数据进行增删改操作,将对象进行导出;

4. 配置管理

(1) 商场配置: 对后台管理系统对应电商站点的商场名称、商场地址、联系电话等信息进行编辑操作;

(2) 运费配置: 配置订单所需运费的相关规则;

(3) 订单配置: 配置订单中的下单超时、发货超期、自动确认收货等相关规则;

5. 统计报表

(1) 用户统计: 对用户增长情况进行统计分析;

(2) 订单统计: 对订单按照数量、用户、价格等方面进行统计分析;

(3) 商品统计: 对商品订单量、下单货物数量、下单货品总额进行统计分析;

2.2 性能需求

本系统所需的性能需求主要包含数据精确度、时间特性和安全性三方面。

1. 对于数据精确度,在进行查找、删除、修改等操作请求,譬如订单查找、商品评论删除的时候,由于其必须保证数据与数据库中的数据相匹配,因此系统应保证响应数据的正确率和覆盖率;

2.对于时间特性,为满足大多数用户的使用要求和体验,数据的响应时间,更新时间,处理时间和运行时间都应控制在2秒以内;

3.对于安全性,本系统作为在线商品销售平台,存在一定安全风险,因此至少需要包含权限控制、重要数据加密、数据备份和记录日志等相关功能或模块。

(1) 权限控制:根据不同用户的角色设置相应的权限,用户的重要操作被相应日志记录以备查看;

(2) 重要数据加密:对用户口令、重要参数等数据根据一定算法进行加密;

(3) 记录日志:可记录系统运行时发生的所有操作,以保证系统出错时可根
据错误记录查找原因。

2.3 可靠性、可用性需求及出错处理需求

1.可靠性需求

本系统作为在线商品销售系统的一部分,涉及商业、交易等内容,因此对可靠性的要求较高。本系统设计时应确保出错率达到标准以下;

2.可用性需求

对于可用性需求,主要集中在以下几点:

(1) 界面要求

本系统经人工操作,因此需保证界面简洁,操作简单,无引发歧义部分;

(2) 规范提示

在系统的某些风险操作,例如删除商品信息时,系统应显示提示框确认删除,以防用户误删数据

3.出错处理需求

本系统如果发生环境错误,应首先能够自动进行出错处理,查找出错部分并上报给管理员。如果系统响应了错误的环境信息导致长时间无法响应操作时,应在用户可以接受的范围内将系统关闭。

2.4 接口需求和约束

1.本系统的接口需求主要包括用户界面、硬件接口和软件接口等方面内容。

(1) 用户界面

本系统对用户提供的用户界面,对于用户来说日常操作均在用户界面中完成。用户界面可分为用户登录界面、功能选择界面和功能操作界面。用户登录界面可以进行用户合法登录的鉴权,功能选择界面为用户提供相关功能的分区,功能操作界面

是用户进行某项操作时的具体界面。

(2) 硬件接口

表 2-1 硬件接口需求

处理器要求	Intel Xeon E5450 以上
内存要求	4GB 以上
其他要求	建议提升硬盘驱动器性能

(3) 软件接口

表 2-2 软件接口需求

操作系统平台	Windows Server 2008 或 Linux 发行版
运行环境	MySQL、nginx

2.约束

表 2-3 环境约束

数据库平台	MySQL
开发语言	JAVA
开发工具和库	IDEA+Spring Boot、Vue

2.5 将来可能出现的需求

本系统是面向商家的在线商城后台系统，未来可能会开发与后台管理系统相匹配的、面向消费者的前台购物商城系统。因此，需设计好消费者客户端与后台管理系统沟通的相应 API 接口，遵循《设计模式》课程中的“面向修改关闭，面向扩展开放”的原则，通过预留等形式，尽可能降低需求扩展所产生的成本或代价。

3 数据库设计

3.1 数据表结构

3.1.1 用户相关

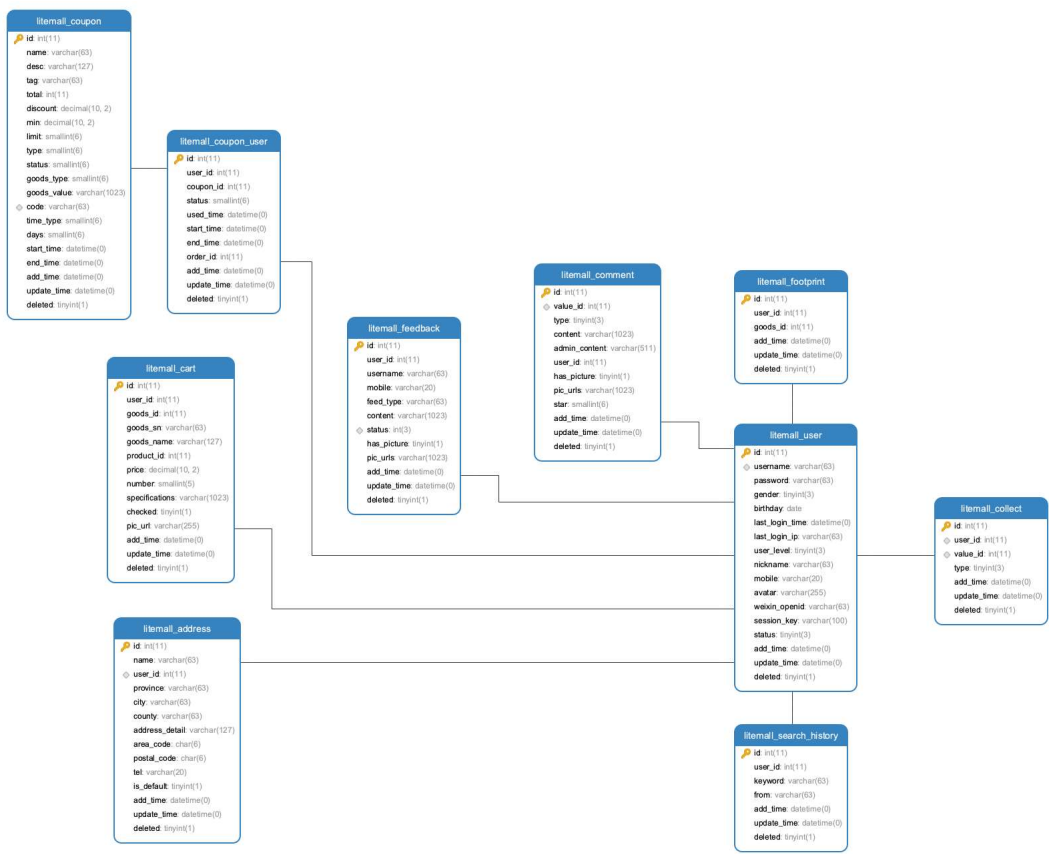


图 3-1 用户相关数据表结构

3.1.2 商品相关

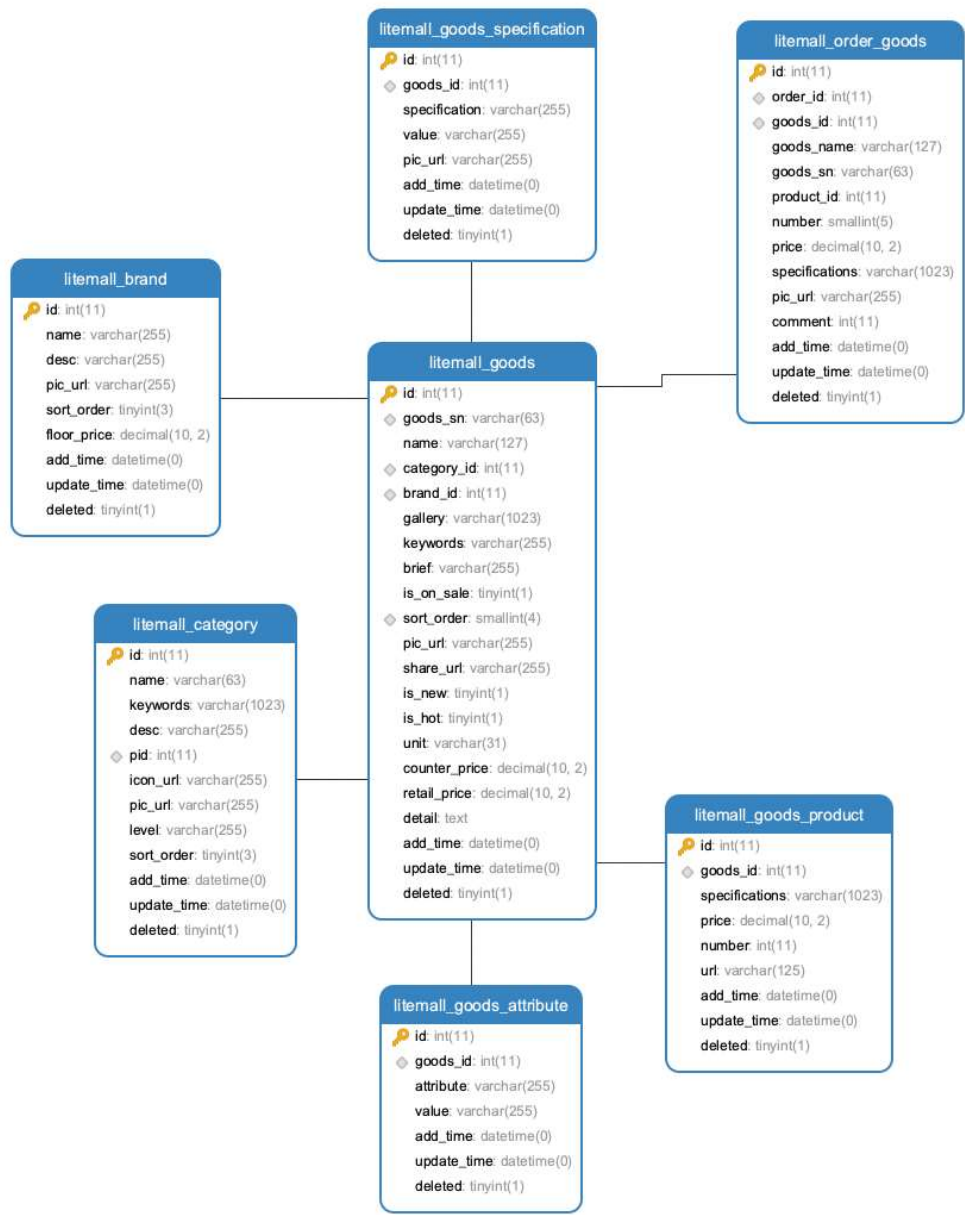


图 3-2 商品相关数据表结构

3.1.3 订单相关



图 3-3 订单相关数据表结构

3.1.4 其他

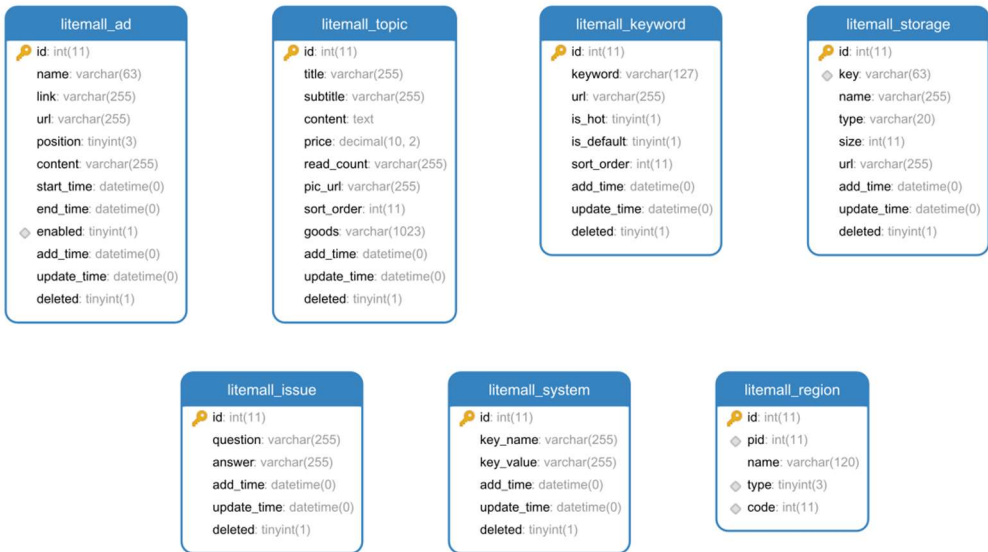


图 3-4 其他数据表结构

3.2 数据库设计

3.2.1 商品和货品设计

这里商品存在商品表（mall_goods），商品属性表（mall_goods_attribute），商品规格表（mall_goods_specification），商品货品表（mall_goods_product）四种表。

商品表是一种商品的基本信息，主要包括商品介绍，商品图片，商品所属类目，商品品牌商等；

商品属性表其实也是商品的基本信息，但是由于是一对多关系，因此不能直接保存在商品表中，因此采用独立的商品参数表，通常是商品的一些公共基本商品参数；

商品规格表是商品进一步区分货品的标识，例如同样一款衣服，基本信息一致，基本属性一致，但是在尺寸这个属性上可以 把衣服区分成多个货品，而且造成对应的数量和价格不一致。商品规格可以看着是商品属性，但具有特殊特征。

商品规格和规格值存在以下几种关系：

单一规格和单一规格值，最常见的，即当前商品存在一种货品；

单一规格和多个规格值，较常见，即当前商品基于某个规格存在多种货品，通常价格都是相同的，当然也可能不相同；

多个规格和单一规格值，可以简化成第一种情况，或者采用第四种情况，通常实际情况下不常见；

多个规格和多个规格值，通常是两种规格或者三种规格较为常见，而且对应的价格不完全相同。

商品货品表则是最终实现商品库存管理、购买业务的实体对象，存在多个规格值、数量和价格。 例如，同样的衣服品牌，可能因为不能尺寸和颜色而存在最终的货品，这里每个货品的价格可以一样，也可以不一样。

总结一下，一个普通商品，实际上在数据库中，存在一个商品表项，存在（至少 0 个）多个商品属性表项目，存在（至少一个）多个商品规格表项， 存在（至少一个）多个货品表项。

3.2.2 订单设计

订单信息主要由基本信息、商品信息、地址信息、费用信息、快递信息、支付信息和其他信息组成，由 mall_order 表和 mall_order_goods 表保存。

表 3-1 订单信息表

信息类型	定义
基本信息	订单创建时的一些基本信息，例如用户、订单状态和订单留言等。 其中订单状态是最重要的信息
商品信息	由于订单可以存在多个商品，因此订单的商品信息是由独立的订单商品表记录（可能更应该称为货品）
费用信息	订单一些费用情况，例如商品总价、优惠减免和实际付费等
收货信息	用户下单时选择的收货地址以及联系人信息
快递信息	目前快递信息仅仅记录快递公司、快递单号、快递发出时间。 而如果快递过程中如果存在一些异常，例如物品丢失，则目前系统难以处理
支付信息	订单商品的评论情况
其他信息	其他相关信息

3.2.3.1 订单状态

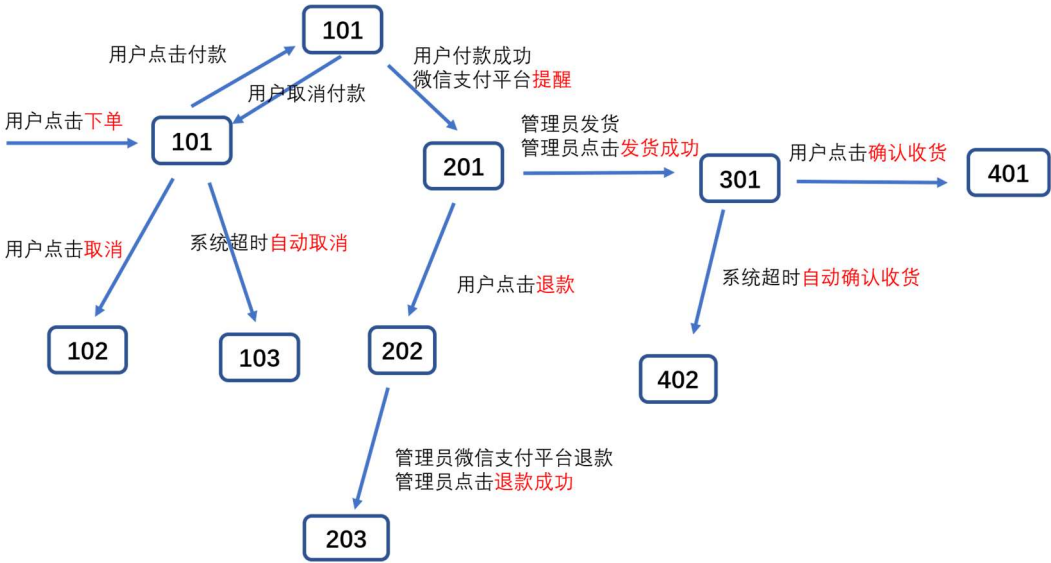


图 3-5 订单状态图

订单分成几种基本的状态：

表 3-2 订单状态表

订单状态码	对应状态
101	此时订单生成，记录订单编号、收货地址信息、订单商品信息和订单相关费用信息

201	状态码 201，此时用户微信支付付款，系统记录微信支付订单号、支付时间、支付状态
301	状态码 301，此时商场已经发货，系统记录快递公司、快递单号、快递发送时间。当快递公司反馈用户签收后，系统记录快递到达时间
401	状态码 401，当用户收到货以后点击确认收货，系统记录确认时间。
以上是一个订单成功完成的基本流程，但实际中还存在其他情况	
102	用户下单后未付款之前，点击取消按钮，系统记录结束时间
103	用户下单后半小时内未付款则系统自动取消，系统记录结束时间
202	用户付款以后未发货前，点击退款按钮，系统进行设置退款状态，等待管理员退款操作
203	管理员在管理后台看到用户的退款申请，点击退款按钮进行退款操作
402	用户已签收却不点击确认收货，超期 7 天以后，则系统自动确认收货。用户不能再点击确认收货按钮，但是可以评价订单商品

此外，当订单状态码是 102、103、203、401 和 402 时，订单可以执行删除操作。目前的设计是不执行物理删除，而是逻辑删除，因此用户查看自己订单时将看不到这些“已删除”的订单。

注意：在上图中可以看到 101 到 101 的状态变化，这里只是小商场用户的操作，不会影响订单状态码。如果用户点击付款时，后端服务会生成预支付会话 id，但是不会影响订单状态。如果而用户支付过程中，放弃支付，则也不会影响订单状态。

3.2.3 用户操作

订单状态码标识了订单的状态，但是对于用户而言，真正关心的只是他们能够进行的操作，目前支持：

支付，如果下单后未立即支付，则订单详情页面会出现支付按钮；

取消，如果用户未支付，则订单详情页面会出现取消按钮；

退款，如果用户支付后但是管理员未发货，则订单详情页面会出现退款按钮；

确认收货，如果管理员已发货，则订单详情页面会出现确认收货按钮；

申请退货，如果用户已经确认收货同时未超过一段时间，则订单详情页面会出现申请退货按钮； 注意，这里如果是系统超时自动确认收货，则不会出现；

去评价，如果用户确认收货以后，则订单详情页面会出现去评价按钮；

再次购买，如果用户确认收货以后，则订单详情页面会出现再次购买按钮；

删除，如果当前订单状态码是 102、103、203、401 和 402 时，则订单详情页面会出现删除订单按钮； 注意，这里的删除操作是逻辑删除，即设置订单的删除状态 deleted。

因此订单状态码和用户操作之间存在映射关系：

表 3-3 订单状态码与用户操作的映射关系

订单状态码	用户操作
101	用户可以支付、取消
102	用户可以删除
103	用户可以删除
201	用户可以退款
203	用户可以删除
301	用户可以确认收货
401	用户可以删除、去评价、申请售后、再次购买
402	用户可以删除、去评价、申请售后、再次购买

3.2.4 申请售后

当用户确认收货或者系统自动确认收货以后，订单可以申请售后。 目前仅支持订单整体售后，而不支持订单商品独立售后。 这是因为：订单存在商品售价、优惠券减免、团购减免以及物流运费属性， 如果要支持单个商品退款，那么存在一个需要解决的问题就是单个商品的 退款金额如何计算。如果开发者这里考虑清楚，也可以参考当前代码实现 订单商品独立售后

mall_order 表中存在 aftersale_status 字段，记录订单售后状态。 而具体的售后记录则是 mall_aftersale 表记录。

这里 type 字段表示当前售后类型，目前存在三种类型：

如果 type=0，即“未收货退款”，通常是系统超时自动确认收货，而实际上用户没有收货，因此可以选择这个；

如果 `type=1`，即“无需退货退款”，通常是用户确认收货后申请售后，而管理员同意可以不需要退货，直接退款给用户；

如果 `type=2`，即“退货退款”，通常是用户确认收货后申请售后，管理员同意用户退货，当管理员收到货以后再退款给用户。

需要注意的是：当前实现中，如果是“退货退款”类型，那么管理员在进行退款以后，系统会自动恢复货品数量。这是因为管理员完成“退货退款”售后，说明管理员已经收到用户的退货。开发者可以改变这里的实现逻辑，例如采用独立的退货入库流程。

`status` 字段表示当前售后状态，分别是：

如果 `status=0`，未申请售后；

如果 `status=1`，用户申请售后，等待管理员审核；

如果 `status=2`，管理员审核通过，等待管理员退款；

如果 `status=3`，管理员已退款，售后完成；

如果 `status=4`，管理员审核不通过，售后完成；

如果 `status=5`，用户已取消售后，当用户在申请售后以后可以在管理员审核前申请取消。

这里需要补充的是：订单 `mall_order` 表的 `aftersale_status` 字段，和订单售后 `mall_aftersale` 表的 `status` 字段是完全一致的，方便前端分别查询订单状态和订单售后状态。

`amount` 字段表示当前售后退款金额，正如前面所述当前仅支持订单整体售后，因此目前设计的退款金额是 订单实际付款-订单运费。

3.2.5 商品评价

在 `mall_order` 表中存在 `comments` 字段，表示有几个订单商品没有评价；而在 `mall_order_goods` 表中存在 `comment` 字段，表示当前订单商品的评论 ID。

当用户确认收货以后，`comments` 设置当前订单中未评价的商品数量。而 `comment` 设置 0；

当用户评价一个订单商品，`comments` 会减一，而 `comment` 指向新创建的评论；

如果用户不评论超期，`comments` 会设置 0，而 `comment` 设置 -1；

3.2.6.1 评论设计

评论表 `mall_comment` 保存评论相关的信息，其中最关键的是 `type` 字段和 `value_id` 字段。

这里 `type` 字段表示当前评论类型，目前存在两种类型：

如果 type=0，则当前评论是订单商品评论，value_id 是订单商品 ID；

如果 type=1，则当前评论是专题评论，value_id 是专题 ID；

admin_content 字段则拥有记录管理后台管理员对用户评论的回复。

3.2.6 系统配置设计

系统配置表 mall_system 保存系统的配置信息。在 Java 代码层系统配置表只能执行更新操作，不能执行创建和删除操作。也就是说，系统配置数据都应该是开发者基于系统的配置需求在数据库中手动创建。

3.2.7 存储对象设计

存储对象表 mall_storage 保存上传文件信息。

当用户或者管理员上传图像时，图像文件会保存到本地或者第三方云存储服务中，同时在存储对象表中记录一下。

3.2.8 操作日志设计

业务日志表 mall_log 记录管理员的关键性操作。

本项目采用在方法内手写业务日志代码方式记录业务操作行为及结果。虽然比较繁琐，但是可以保证记录是细颗粒的。而且，如果管理员最终关心的操作较少，那么实际上需要写的代码不是很多。

考虑到语义，操作业务应该是“谁做了什么操作，结果成功还是失败，失败原因是什么，补充信息是什么”，因此这里设计的业务日志表关键字段如下：管理员、IP 地址、操作分类、操作动作、操作状态、操作结果、补充信息

3.2.11.1 操作类别

这里的日志类型设计成四种（当然开发者需要可以扩展）

一般日志：用户觉得需要查看的一般操作日志，建议是默认的日志级别

安全日志：用户安全相关的操作日志，例如登录、删除管理员

订单日志：用户交易相关的操作日志，例如订单发货、退款

其他日志：如果以上三种不合适，可以选择其他日志，建议是优先级最低的日志级别

当然建议开发者应该和最终用户讨论交流，记录真正关键性的业务操作，例如登录相关或订单相关等。

3.2.11.2 操作结果

如果操作成功，可以使用操作结果字段记录被操作的对象。当然，有些操作没有具体对象，那么可以省略。

如果操作失败，也可以使用操作结果字段记录失败的原因。

3.2.11.3 操作失败

虽然这里有操作状态字段和操作结果字段，可以记录操作失败的状态。但是通常失败操作不会对系统或者数据库带来影响，因此实际上开发者其实不需要记录太多操作失败的日志，而是记录操作成功的日志，告诉系统管理员当前状态的变化。

当然，是否记录操作失败取决于开发者或者最终用户是否需要。例如，登录这里应该记录用户登录失败的日志，因为保存的 IP 地址可以帮助管理员了解系统被访问的情况。

3.2.9 通用设计

除了以上表，数据库还存在其他一些业务表，例如专题 `mall_topic`，以下是一些表设计中无具体业务意义可通用的字段。

3.2.12.1 deleted

除极少数表，其他所有表都存在 `deleted` 字段，支持逻辑删除。因此目前删除数据时，不会直接删除数据，而是修改 `deleted` 字段。当然，数据库管理员可以连接到数据库直接删除数据，或者开发者可以修改这里的逻辑采用物理删除。

3.2.12.2 add_time

除极少数表，其他所有表都存在 `add_time` 字段，记录数据创建时间。

3.2.12.3 update_time

除极少数表，其他所有表都存在 `update_time` 字段，记录数据修改时间。

此外，可以利用 `update_time` 来实现乐观锁更新机制。

4 系统设计与实现

4.1 系统架构

4.1.1 简介

mall 是一个简单的商场系统，基于现有的开源项目，重新实现一个完整的后端项目的网页管理端。

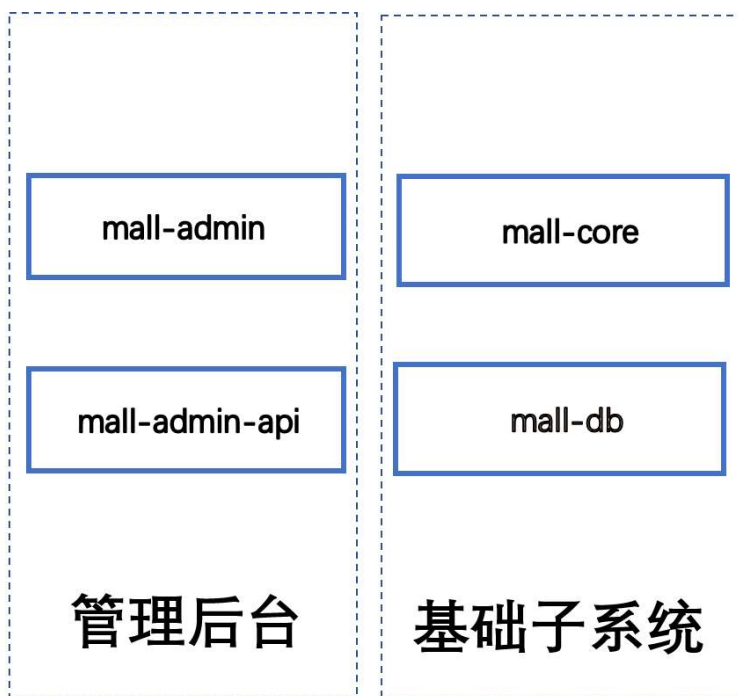


图 4-1 mall 系统架构图

项目的架构是两个系统和三个模块：

(1) 基础系统子系统(platform): 由数据库、mall-core 模块和 mall-db 模块组成;

(2) 管理后台子系统(admin): 由 mall-admin-api 模块和 mall-admin 模块组成。

模块的开发涉及到两种技术栈：

(1) Spring Boot 技术栈: 采用 IDEA 开发工具，开发 mall-core、mall-db、mall-admin-api 模块;

(2) Vue 技术栈: 采用 VSC 开发工具，开发 mall-admin 模块。

4.1.1.1 项目特点

项目存在以下特点：

(1) 数据库方面，只是简单的表，表和表之间的依赖关系没有采用外键设计，

而是依赖 Java 代码在 service 层面或者业务层面保证。这样做的好处是数据库频繁改动很方便，不会因为外键而导致数据库难以修改；

- (2) 每种技术栈仅涉及最基础的技术；
- (3) 安全方面，仅采用最基本的代码，提供简单基本的安全服务；
- (4) 性能方面，没有涉及内存数据库缓存功能，而是完全依赖 MySQL；
- (5) 对象存储服务方面，支持本地存储和第三方云存储方案。
- (6) 消息通知方面，支持邮件通知、第三方云短信通知和微信模板通知；
- (7) 部署方便，支持多服务部署和一键部署脚本；
- (8) 文档全面，虽然还在开发中，但是规划中文档和代码注释一定会完成，帮助开发者理解项目。

总之，目前的系统只是为了学习技术和业务而开发的一个简单商场原型系统。虽然缺失很多企业级功能，但是是完整和合理的原型系统。

4.1.2 系统功能

- (1) 会员管理：会员管理、收货地址、会员收藏、会员足迹、搜索历史、意见反馈
- (2) 商城管理：行政区域、品牌制造商、订单管理、商品类目、通用问题、关键词、渠道管理（待定）
- (3) 商品管理：商品列表、商品上架、商品编辑、用户评论
- (4) 推广管理：广告管理、专题管理、团购规则、团购活动
- (5) 系统管理：管理员、通知管理、对象存储、权限管理、定时任务（待定）、操作日志
- (6) 配置管理：商场配置、小程序配置、运费配置、订单配置
- (7) 统计报表：用户统计、订单统计、商品统计
- (8) 个人：通知中心、密码修改

4.1.3 项目技术

4.1.3.1 技术参考

1.Spring Boot 技术

Spring Boot 技术栈参考以下文档或者项目：

- (1) MySQL

了解创建数据库和表、添加、查询、更新和删除即可。

(2) Spring Boot 2.x

<https://docs.spring.io/spring-boot/docs/2.1.5.RELEASE/reference/htmlsingle/#getting-started-introducing-spring-boot>

<https://docs.spring.io/spring-boot/docs/2.1.5.RELEASE/reference/htmlsingle/#using-boot-maven>

这里需要了解 RestController, Service 等注解, 以及如何使用自动化配置。Spring Boot 支持很多功能, 开发者使用时查阅。

(3) Mybatis

<http://www.mybatis.org/mybatis-3/>

<http://www.mybatis.org/mybatis-3/java-api.html>

<http://www.mybatis.org/mybatis-3/sqlmap-xml.html>

这里可以简单了解, 而使用 Mybatis Generator 来生成 Java 代码使用即可。

(4) Mybatis Generator

<http://www.mybatis.org/generator/running/runningWithMaven.html>

<http://www.mybatis.org/generator/generatedobjects/results.html>

<http://www.mybatis.org/generator/generatedobjects/exampleClassUsage.html>

(5) Mybatis PageHelper

<https://github.com/pagehelper/Mybatis-PageHelper/blob/master/wikis/en/HowToUse.md>

2. Vue 技术

(1) Vue

<https://cn.vuejs.org/index.html>

(2) Vant

<https://youzan.github.io/vant/#/zh-CN/intro>

(3) Element

<https://element.eleme.cn/#/zh-CN/component/installation>

(4) vue-element-admin

<https://github.com/PanJiaChen/vue-element-admin>

<https://panjiachen.github.io/vue-element-admin-site/zh/>

4.1.3.2 项目阶段

接下来, 从项目的开发、部署(测试)和上线三个阶段介绍 mall。

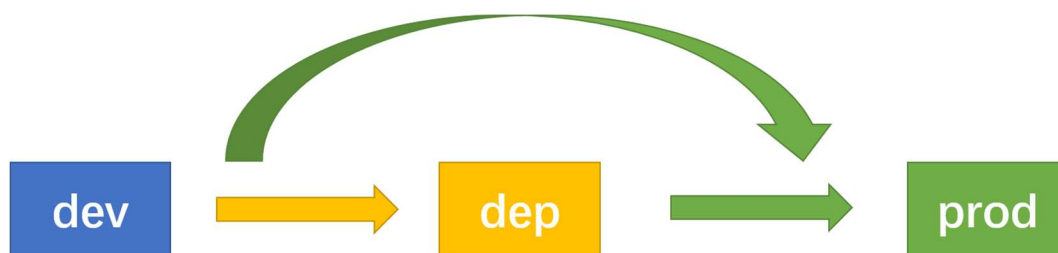


图 4-2 mall 项目阶段图

首先需要明确的是三个不同阶段：

(1) dev

即 develop 或者 development, 这里指开发阶段, 通常代码是直接在本机编译、运行和测试。此外, 这里服务访问地址通常是 localhost。这里的“用户”主要是指开发者本身。

(2) dep

即 deploy 或者 deployment, 这里指部署 (测试阶段), 通常代码已经编译打包运行在远程服务器中, 可以对外服务。此外, 这里服务访问地址通常是 IP 地址。如果 IP 是公网 IP, 那么部署以后就可以对外服务; 如果是内网地址, 那么只能内网访问。这里的“用户”主要是指开发者本身、测试者; 当然, 如果是局域网或者不介意 IP 访问的, 那么这里的“用户”也可能是最终使用者用户。

(3) prod

即 product 或者 production, 这里指上线阶段, 通常也是代码编译打包运行在远处服务器中可以对外服务。此外, 这里服务访问地址通常是域名地址, 同时端口是 80web 端口。上线以后直接面向的是最终用户。虽然服务的代码本身和 dep 是完全一样的, 但是考虑到场景的不同, 上线阶段可能在运行环境方面需要做调整, 例如采用反向代理屏蔽内部实际项目结构。此外, 最大的不同应该是上线环境下要使用域名和 80 端口, 而部署阶段则更为自由。

其次, 需要明确的是, 这里划分三种阶段不是简单的文档说明, 还直接影响项目本身的行为和代码编译结果, 因此开发者需要清晰的了解;

最后, 其实 dep 和 prod 不存在先后关系。例如, 如果开发者已经存在域名和生产环境, 可以直接跳过 dep 阶段, 而直接部署在线上环境中。因此有些时候, 这里部署和上线是一个阶段。

当然, 这里仍然建议先 dep 后 prod, 是因为对于第一次开发而言, 先 dep 阶段可以避免对域名、https 证书等非业务相关工作的干扰。

此外, 有些业务功能 (例如微信支付) 必须是域名访问, 那么开发和部署阶段

可以先采用模拟或跳过的形式，先不开开发和测试这样业务功能，等其他功能开发完毕和部署测试成功以后，再来开发这些线上环境才能运行的功能，此时会有一个好的基础。

4.1.4 开发方案

4.1.4.1 数据库

数据库环境设置过程如下：

(1) 安装 MySQL;

(2) 创建数据库、用户权限、数据库表和测试数据;数据库文件存放在 mall-db/sql 文件夹中，请开发者在 MySQL 中按照顺序运行以下脚本：

mall_schema.sql，用于创建数据库、用户和权限；

mall_table.sql，用于创建表；

mall_data.sql，用于导入测试数据。

注意：建议采用命令行或者 MySQL Workbench。如果采用 Navicat 可能导入失败。

如果开发者运行 mall_schema.sql 失败，可以打开该文件：

```
drop database if exists mall;
drop user if exists 'mall'@'%';
create database mall default character set utf8mb4 collate utf8mb4_unicode_ci;
use mall;
create user 'mall'@'%' identified by 'mall123456';
grant all privileges on mall.* to 'mall'@'%';
flush privilege
```

可以看到几个命令，用于创建数据库、用户和访问权限，因此开发者可以利用命令或者工具完成这里的功能即可。

4.1.4.2 Spring Boot 开发环境

(1) 安装 JDK8（可以是 Oracle JDK 或者 OpenJDK）

(2) 安装 Maven

(3) 安装 Git（可选）

(4) 安装 IDEA Community，建议安装 Maven 插件和 Git 插件。这里 IDEA 社区版即可，不要求 IDEA 商业版。Eclipse 没有试过，但应该也是可行的。

(5) IDEA 导入本项目

(6) 采用 Maven 命令安装依赖库

例如：

```
cd mall
mvn install
```

或者采用 IDEA 的 Maven 插件安装本项目依赖库，点击 install

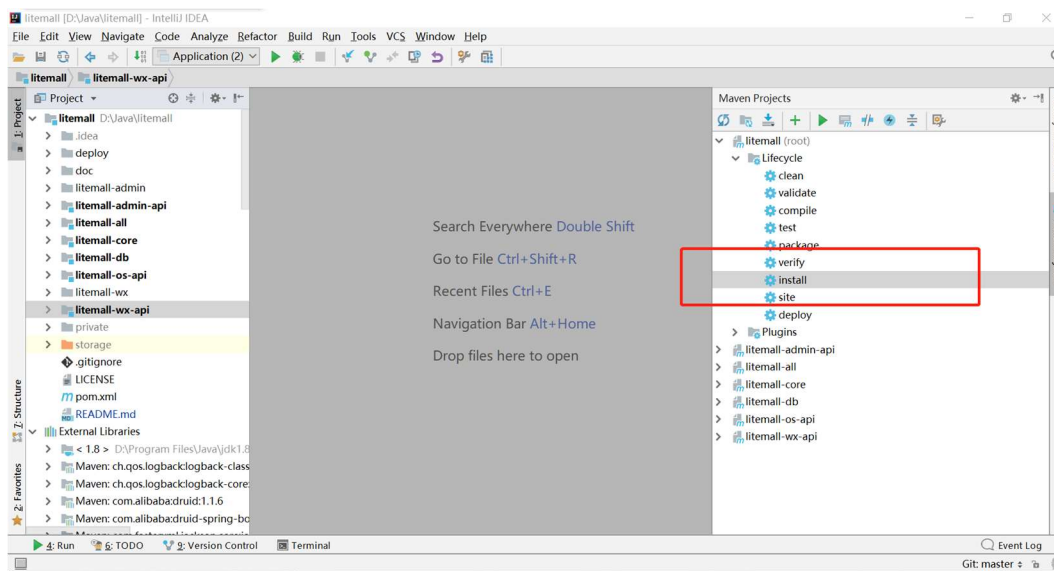


图 4-3 Spring Boot 环境安装

(7) 采用 Maven 命令编译本项目

例如：

```
cd mall
mvn compile
```

或者采用 IDEA 的 Maven 插件编译本项目，点击 compile

打开浏览器，输入 <http://localhost:8080/wx/index/index>
<http://localhost:8080/admin/index/index> 如果出现 JSON 数据，则模块运行正常。

4.1.4.3 Vue 开发环境

- (1) 安装 nodejs
- (2) 安装依赖库

```
cd mall/mall-admin
npm install -g cnpm --registry=https://registry.npm.taobao.org
cnpm install
```

- (3) 编译并运行

```
cnpm run dev
```

然后，打开浏览器，输入 <http://localhost:9527>。如果出现管理后台登录页面，

则表明管理后台的前端运行正常；

(4) 然后点击登录，如果能够成功登录，则表明管理后台的前端和后端对接成功，运行正常。

本项目采用 VSC (Visual Studio Code) 开发 mall-admin 模块，开发者也可以采用其他熟悉的 IDE。

4.1.5 部署方案

上一节中介绍的是开发阶段时一些关键性开发流程。本节将介绍代码开发成功以后开始部署项目时一些关键性流程。

首先，需要明确的是开发时项目使用的服务地址是本地地址，即 localhost；而部署时则应该根据具体情况设置合理的服务器地址和端口。

其次，需要明确的是各模块之间的关系：

- mall-admin-api 模块会包含 mall-core 模块和 mall-db 模块，部署在服务器中
- mall-admin 编译出的静态文件放在 web 服务器或者 tomcat 服务器，此外服务器地址设置指向 3 中 mall-admin-api 所在地址

最后，如果项目部署云服务器，则根据开发者的部署环境在以下文件中或代码中修改相应的配置。

- (1) MySQL 数据库设置合适的用户名和密码信息；
- (2) 后端服务模块设置合适的配置信息；
- (3) 管理后台前端 mall-admin 模块.env.deployment 中的 VUE_APP_BASE_API 设置管理后台后端服务的服务地址。

实际上，最终的部署方案是灵活的：

- 可以是同一云服务器中安装一个 Spring Boot 服务，同时提供 mall-admin、mall-admin-api 服务
- 可以单一云服务器中仅安装一个 tomcat/nginx 服务器部署 mall-admin 静态页面分发服务，然后部署后端服务；
- 也可以把 mall-admin 静态页面托管第三方 cdn，然后开发者部署后端服务
- 当然，甚至多个服务器，采用集群式并发提供服务。

4.1.6 上线方案

正式环境需要域名和 HTTPS 证书，本节采用 www.example.com 域名作为示例。列举一种单机单服务上线方案，即一个 all 后端服务，同时提供两种数据：提

供管理后台的前端文件；提供管理后台前端所需要的数据；

4.1.6.1 域名

- (1) 注册域名，通常商业性的网站采用.com
- (2) 解析域名到服务器公网 IP，采用 ping 命令查看是否解析成功
- (3) 备案

4.1.6.2 nginx

1.nginx 安装

采用命令

```
sudo apt-get update  
sudo apt-get install nginx
```

有的文档会指出需要防火墙设置，但是腾讯云服务器防火墙默认没有开启。开发者这里自己可以开启设置，或者直接不开启。

打开浏览器，输入以下地址：

<http://www.example.com>

此时，如果看到 nginx 的欢迎页面，则安装成功。

2.https

- (1) 申请证书
- (2) 下载证书

使用 nginx 文件夹下面的密钥文件，例如 1_www.example.com_bundle.crt 和 2_www.example.com.key

- (3) 部署证书到 nginx

可以参考腾讯云证书安装指引把两个密钥文件保存的/etc/nginx 文件夹，然后修改/etc/nginx/nginx.conf 文件：

```
server {  
    listen 443;  
    server_name www.example.com;  
    ssl on;  
    ssl_certificate /etc/nginx/1_www.example.com_bundle.crt;  
    ssl_certificate_key /etc/nginx/2_www.example.com.key;  
    ssl_session_timeout 5m;  
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  
    ssl_ciphers ECDHE-RSA-AES128-GCM-  
    SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;  
    ssl_prefer_server_ciphers on;  
}
```

```
}
```

(4) 重启 nginx

打开浏览器，输入以下地址：

<https://www.example.com>

此时，可以看到 https 协议的 nginx 欢迎页面。

3.反向代理 Spring Boot 后端

修改/etc/nginx/nginx.conf 文件，配置 nginx 静态 web 文件目录

```
server {  
    location / {  
        proxy_pass http://localhost:8080;  
        proxy_set_header    Host      $host;  
        proxy_set_header     X-Real-IP  $remote_addr;  
        proxy_set_header     X-Forwarded-For $proxy_add_x_forwarded_for;  
    }  
}
```

打开浏览器，输入以下地址：

<https://www.example.com/wx/index/index>

<https://www.example.com/admin/index/index>

<https://www.example.com/admin/index/index>

此时，看到后端数据说明反向代理配置成功。

4.全站加密

服务器自动把 http 的请求重定向到 https

```
server {  
    listen 80;  
    server_name www.example.com;  
    rewrite https://$server_name$request_uri? permanent;  
}
```

打开浏览器，输入以下地址：

<http://www.example.com>

总结，经过以上不同方面的配置，nginx 这里最终的配置是如下：

(1) 证书 1_www.example.com_bundle.crt 和 2_www.example.com.key 放置在 /etc/nginx/文件夹内。

(2) 把/etc/nginx/nginx.conf 文件进行修改，具体可以参考本项目的 nginx.conf

(3) 重启 nginx

4.1.6.3 管理后台上线

在上一节“反向代理 Spring Boot 后端”成功以后，其实管理后台已经上线成功，包括管理后台的前端和后端，会同时对外提供管理后台的前端文件和后端数据。当然，这里开发者需要自己的线上环境在以下文件中或代码中修改相应的配置。

(1) MySQL 数据库设置合适的用户名和密码信息；

(2) 管理后台后端服务模块设置合适的配置信息，建议开发者参考 `deploy/mall` 的外部配置文件，这样可以避免开发者对模块内部的开发配置文件造成修改；

(3) 管理后台前端 `mall-admin` 模块 `.env.production` 中的 `VUE_APP_BASE_API` 设置管理后台后端服务的地址。

4.2 基础系统

目前 mall 基础系统由以下部分组成：

- mall-core 模块
- mall-db 模块

mall-db 模块提供数据库访问服务。

mall-core 模块提供通用服务。

4.2.1 mall-db

mall-db 模块是一个普通的 Spring Boot 应用，基于 mybatis 框架实现数据库访问操作，对外提供业务数据访问服务。

此外，mall-db 最终是作为一个类库被其他模块所依赖使用，因此并不对外直接服务，没有使用 Spring MVC 技术。

技术：Spring Boot 2.x、MySQL、Druid、Mybatis、PageHelper、Mybatis Generator、Mybatis Generator 非官方插件 mybatis-generator-plugin

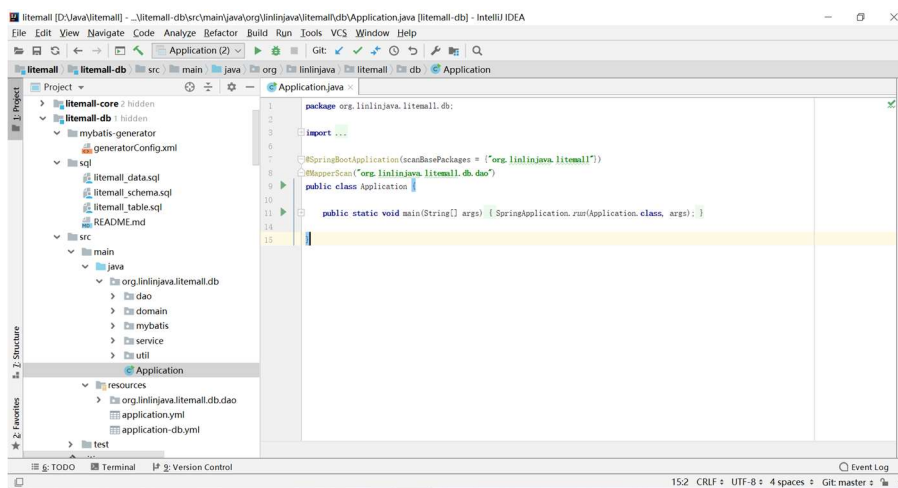


图 4-4 mall-db 模块

这里 mall-db 模块可以分成以下几种代码：mybatis 数据库访问代码、generator 生成代码、非 generator 手动代码、业务代码、mybatis generator 支持代码

(1) mybatis 数据库访问代码

mybatis 数据库访问代码是指 dao 接口代码、dao 数据库 XML 文件和 domain 代码:dao 接口代码，声明了数据库访问接口 dao 数据库 XML 文件，实现了数据库访问操作 domain 代码，则是保存数据库返回数据。

此外，这里的数据库访问代码又进一步分成 mybatis generator 自动生成代码，即基于 mybatis generator 相关插件自动生成上述三种代码或文件；非 mybatis generator 手动代码，则是需要自己编写上述三种代码。

(2) 自动生成代码

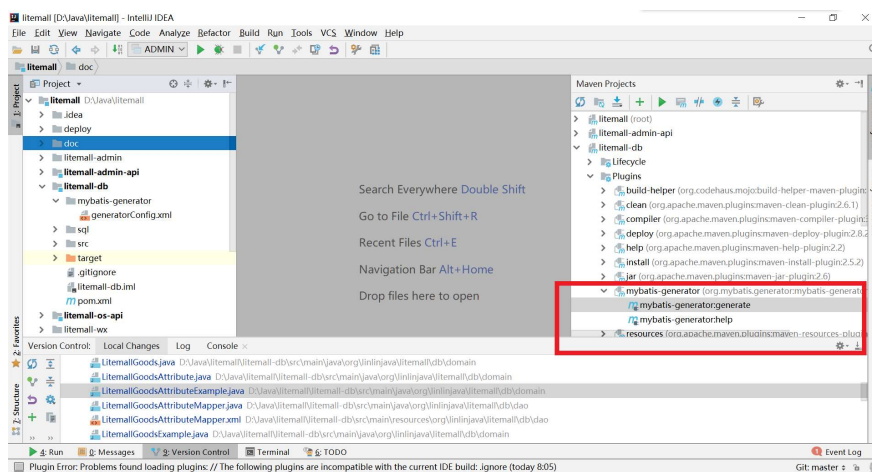


图 4-5 generator 生成代码

如上图所示，双击 mybatis-generator:generate，则 mybatis generator 插件会读取

mybatis-generator 文件夹下的 generatorConfig.xml 文件

根据 jdbcConnection 访问数据库

根据 table，自动生成三种代码：

src 文件夹 org.linlinjava.mall.db.dao 包内的 Java 代码

src 文件夹 org.linlinjava.mall.db.domain 包内的 Java 代码

resources 文件夹 org.linlinjava.mall.db.domain.dao 内的 XML 文件

以上三种代码即可封装对数据库的操作，无需直接操作 sql 代码，而是直接操作 Java 代码来完成对数据库的访问处理。

为了达到数据库访问效率，也可以手动自定义 mapper 文件和对应的 Java 代码。例如，当需要访问两个表的数据时，这里是在业务层通过 Java 代码遍历的形式来访问两个表，也可以通过自定义的 mapper 文件来实现。

接下来，以 mall_brand 表举例说明如何自动生成代码：

mybatis generator 插件会读取 table 标签

```
<generatorConfiguration>
    <table tableName="mall_brand">
        <generatedKey column="id" sqlStatement="MySql" identity="true" />
    </table>
</generatorConfiguration>
```

自动生产 src 文件夹下 domain 包内的 MallBrand.java 类、MallBrandExample.java 类、dao 包内的 MallBrandMapper.java 接口和 resources 文件夹下 dao 包内的 MallBrandMapper.xml 文件。

手动在 service 包内创建 MallBrandService.java 来对外提供具体的服务。例如，为了得到 Brand 列表，那么创建 list 方法，基于前面创建的三个 Java 来实现。

```
@Service
public class MallBrandService {
    @Resource
    private MallBrandMapper brandMapper;

    public List<MallBrand> query(int offset, int limit) {
        MallBrandExample example = new MallBrandExample();
        example.or().andDeletedEqualTo(false);
        PageHelper.startPage(offset, limit);
        return brandMapper.selectByExample(example);
    }
}
```

```
}
```

关于 mybatis generator 的用法，可以自行查阅官网或文档。

(3) 手动代码

虽然 mybatis generator 可以自动生产代码，帮助简化开发工作，但是在涉及到多表操作或特殊数据库操作时，仍然需要自己手动编写基于 mybatis 框架的相关代码。

这里以统计功能举例说明：

dao 代码：在 src 文件夹 org.linlinjava.mall.db.domain 包内的 StatMapper.java 代码定义了数据库访问的接口

domain 代码：如果希望数据库操作返回数据模型，可以在 src 文件夹 org.linlinjava.mall.db.domain 包内创建相应代码。而这里统计功能是采用简化的 List<Map>保存数据，没有定义 domain 代码。

XML 文件：在 resources 文件夹 org.linlinjava.mall.db.domain.dao 内的 StatMapper.xml 文件则是实现真正的数据库访问操作。

service 代码：这里可以在 org.linlinjava.mall.db.service 内定义一个 StatService.java 代码，调用底层 mapper 代码，对外服务。

```
`` @Service public class StatService { @Resource private StatMapper
statMapper;

    public List<Map> statUser() {
        return statMapper.statUser();
    }

    public List<Map> statOrder(){
        return statMapper.statOrder();
    }

    public List<Map> statGoods(){
        return statMapper.statGoods();
    }
} ``
```

(4) 业务代码

虽然所述代码已经能够提供数据库访问操作，但是这里需要进一步地抽象出业务访问层代码，即基于所述代码和实际业务需求实现一些具体业务相关的操作，对其他模块提供便捷业务数据服务。

这里的业务代码往往是单表相关的业务代码，而涉及到多表操作的 java 代码通常是在其他高层模块中实现。这里的业务分层并不是绝对的。例如，可以取消这里的业务代码，而在其他模块中直接调用上述代码。

通常业务层代码在 src 文件夹 org.linlinjava.mall.db.service 包中。

(5) mybatis generator 支持代码

mybatis generator 自动生成代码时，通过内置类型转换器自动把数据库类型转换成 Java 类。例如数据库类型 varchar 自动转化成 java.lang.String。

为了简化数据表的设计，某些字段采用 varchar 来存储 Json 格式的数据。例如商品的图片列表可以直接采用[url0,url1,...]来存储，而不需要设计一个专门商品图片表。

这里通过自定义 TypeHandler，可以实现 Java 的 String[]和数据库类型 varchar 的自动转换。

目前只实现了两个自定义 TypeHandler： * JsonStringArrayTypeHandler 类，实现 String[]和 varchar 的转换，保存的 JSON 数据格式是[string0, string1, ...] * JsonIntegerArrayTypeHandler 类，实现 Integer[]和 varchar 的转换，保存的 JSON 数据格式是[integer0, integer1, ...]

4.2.2 mall-core

mall-core 模块是本项目通用的代码：

config: 通用配置，例如开启 Spring Boot 异步功能。

Util: 工具代码。

Qcode: 本项目定制的分发二维码图片。

Storage: 存储功能，支持本地存储、腾讯云存储、阿里云存储和七牛云存储。

Notify: 通知提醒功能，支持邮件通知、短信通知和微信通知。

Express: 物流服务，查询订单物流信息。

System: 通过 mall-db 模块的数据库访问，读取本项目系统配置信息。

Validator: 提供两个校验注解，帮助后端验证请求参数。

4.2.2.1 config

CorsConfig: 目前开发过程中，CORS 配置是允许所有请求。真正部署时，需要做一些调整，来保证当前的服务只接受来自所设置域名的请求。

GlobalExceptionHandler: 如果系统内部产生了异常而 没有 catch，那么异常的内容会发送到前端。这里通过提供全局异常处理器，来处理所有 没有处理的异常，返回 “系统内部错误” 之类的信息给前端从而达到保护系统的效果。

JacksonConfig: Jackson 做一些设置。

4.2.2.2 util

这里的 util 代码不会涉及具体业务, 例如 mall-db 模块中存在一个 OrderUtil 类处理数据库中 mall_order 表的一些转换工作。

ResponseUtil: 这里是用于设置 response 中 body 的内容格式。

如果是成功则是 :

```
{
    errno: 0,
    errmsg: '成功',
    data: XXX
}
```

如果失败则是:

```
{
    errno: 非 0 的 XXX,
    errmsg: XXX
}
```

JacksonUtil: 当请求时 POST 时, 请求的 json 内容在 body。通常存在存在两种方式取出数据: 如果 json 内容正好对应一个 POJO, 那么在方法中使用 POJO 时, spring 会自动解析填充数据; 或者自己采用 jackson 或者其他 json 处理库手动解析数据。这里 JacksonUtil 简化解析工作。这里代码有局限性, 请谨慎使用, 或者熟悉 Jackson 使用的欢迎优化代码。

CharUtil: 生成固定长度的随机字母字符串或者随机数字字符串。

bcrypt: 这里是用于对用户密码或者管理员密码加密存储。bcrypt 代码本质上是 spring 里面的代码。

4.2.2.3 二维码

见 QCodeService 类。

4.2.2.4 对象存储

对象存储服务目前的目标是支持图片的上传下载。

对象存储服务会自动读取配置配置, 然后实例化服务。

对象存储接口:

```
public interface Storage {
    void store(InputStream inputStream, long contentLength, String contentType,
String keyName);
    Stream<Path> loadAll();
}
```

```
Path load(String keyName);
Resource loadAsResource(String keyName);
void delete(String keyName);
String generateUrl(String keyName);
}
```

本地存储服务：见 `LocalStorage` 类。

腾讯云存储服务：见 `TencentStorage` 类。

阿里云存储服务：见 `AliyunStorage` 类。

七牛云存储服务：见 `QiniuStorage` 类。

4.2.2.5 消息通知

消息通知用于通知用户或者管理员。

邮件通知：见 `NotifyService` 类的 `notifyMail` 方法。

短信通知：见 `NotifyService` 类的 `notifySms` 和 `notifySmsTemplate` 方法。而短信通知实现类见 `TencentSmsSender` 类。也就是目前仅支持腾讯云短信服务，其他短信服务不支持。此外，必须先先在腾讯云短信平台申请模板才能使用。

微信通知：见 `NotifyService` 类的 `notifySms` 和 `notifyWxTemplate` 方法。而微信通知实现类见 `WxTemplateSender` 类。必须在微信平台申请模板才能使用。

4.2.2.6 物流跟踪

物流跟踪是基于第三方服务快鸟物流查询服务。需要申请才能使用。见 `ExpressService` 类。

4.2.2.7 校验注解

自定了两个校验注解，帮助校验 HTTP 参数。

(1) `Order`：校验用户请求参数值只能是 `desc` 或者 `asc`。注意，这里的 `Order` 不是订单的意思，而是排序的意思。

(2) `Sort`：校验用户请求参数值只能是 `add_time` 或者 `id`

4.3 管理后台

项目技术架构：

(1) 管理后台前端，即 `mall-admin` 模块

vue

vuex

vue-router

axios

element

vue-element-admin 4.2.1

(2) 管理后台后端, 即 mall-admin-api 模块

Spring Boot 2.x

Spring MVC

4.3.1 mall-admin-api

4.3.1.1 授权服务

见 AdminAuthController 类

4.3.1.2 用户管理服务

用户管理服务进一步分成:

会员管理服务, 见 AdminUserController 类

收货地址服务, 见 AdminAddressController 类

会员收藏服务, 见 AdminCollectController 类

会员足迹服务, 见 AdminFootprintController 类

搜索历史服务, 见 AdminHistoryController 类

意见反馈服务, 见 AdminFeedbackController 类

4.3.1.3 商场管理服务

商城管理服务进一步分成:

行政区域服务, 见 AdminRegionController 类

品牌制造商服务, 见 AdminBrandController 类

商品类目服务, 见 AdminCategoryController 类

订单管理服务, 见 AdminOrderController 类

通用问题服务, 见 AdminIssueController 类

关键词服务, 见 AdminKeywordController 类

4.3.1.4 商品管理服务

商品服务, 见 AdminAdminController 类

4.3.1.5 系统管理服务

系统管理服务进一步分成:

管理员服务, 见 AdminAdminController 类

对象存储服务, 见 AdminStorageController 类

4.3.1.6 其他服务

统计服务, 见 AdminStatController 类

个人服务，见 AdminProfileController 类

4.3.1.7 权限控制

后端实现权限管理功能来支持一定的安全控制。

4.3.1.8 权限管理

1.权限设计

权限控制在数据库层面涉及到三个表 mall_admin, mall_role 和 mall_permission:

mall_admin 表中存在 roleId 字段，保存角色 ID 数组；

mall_role 表记录角色名称和角色介绍；

mall_permission 表记录角色所用于的权限值。

通常是五个表，user, role, user_role, permission, role_permission。但是本项目只需要三个表：user_role 表的关联关系，可以通过 user 表的 roles 字段完成，因此可以省略；permission 表省略。

本项目对管理员而言只有一种权限，但是这个权限本身对应了菜单权限元数据、按钮权限元数据、接口权限元数据。因此，管理员勾选一个权限以后，后台权限即授权成功，同时前端的菜单权限和按钮权限也自动调整。

后端权限基于 shiro 来实现，相关代码见 mall-admin-api 模块。

2.基本配置

config 子包的 ShiroConfig 引入了 Shiro 并配置了 shirFilter、realm 和 sessionManager;

shiroFilter 配置只允许少量 url 可以匿名访问，其他 url 都需要登录才能访问；

realm 设置的是 shiro 子包的 AdminAuthorizingRealm 类，该类作用是认证和授权的功能；

sessionManager 设置的是 shiro 子包的 AdminWebSessionManager 类，该类作用是重置会话管理器。默认会话管理器是基于 cookie 实现会话保持，而这里是基于自定义头部实现会话保持。

经过以上步骤，shiro 就配置正常。当管理员登录时，先认证；认证成功，则授权，在后端内保存 roles 和 permissions；同时，在响应头部写入自定义头部和 sessionId；认证失败，则抛出认证异常；管理员再次访问页面时，shiro 通过自定义头自动认证成功。

3.权限校验

但是以上只完成授权功能，而没有完成权限校验功能。这里应该在 Controller 类的方法中使用 RequiresPermissions 注解。例如：

```
@RestController
```

```

@RequestMapping("/admin/ad")
public class AdminAdController {

    @RequiresPermissions("admin:ad:list")
    @RequiresPermissionsDesc(menu={"推广管理", "广告管理"}, button="查询")

    @GetMapping("/list")
    public Object list(String name, String content,
                      @RequestParam(defaultValue = "1") Integer page,
                      @RequestParam(defaultValue = "10") Integer limit,
                      @Sort @RequestParam(defaultValue = "add_time")
String sort,
                      @Order @RequestParam(defaultValue = "desc")
String order) {
        List<MallAd> adList = adService.querySelective(name, content, page,
limit, sort, order);
        int total = adService.countSelective(name, content, page, limit, sort,
order);

        Map<String, Object> data = new HashMap<>();
        data.put("total", total);
        data.put("items", adList);

        return ResponseUtil.ok(data);
    }
}

```

而具体的权限校验逻辑则由 shiro 自动完成：

- (1) 登录成功或者会话登录，shiro 已经有当前用户的权限列表；
- (2) 访问权限保护的方法时，shiro 通过 RequiresPermissions 注解得到所需操作权限列表；
- (3) 测试已分配的权限和操作所需权限是否一致，如果一致则可以调用方法，否则抛出无权限的异常。

4. 权限描述

在 annotation 子包中存在一个自定义的 RequiresPermissionsDesc 注解

```

@Target({ElementType.TYPE, ElementType.METHOD})
@Retention(RetentionPolicy.RUNTIME)

```

```
public @interface RequiresPermissionsDesc {  
    String[] menu();  
    String button();  
}
```

本项目通过搜索 `RequiresPermissions` 注解和 `RequiresPermissionsDesc` 注解（见 `util` 子包的 `PermissionUtil` 类），在内部生产了 `Permission` 类的集合：

```
public class Permission {  
    private RequiresPermissions requiresPermissions;  
    private RequiresPermissionsDesc requiresPermissionsDesc;  
    private String api;  
}
```

也就是说，对于一个权限，既有权限值（`requiresPermissions` 的 `value`），也有权限对应的操作 API，也有权限的描述内容（`requiresPermissionsDesc` 的 `menu` 和 `button`）。

这里利用注解方式可以不需要在数据库中保存权限信息，但是在灵活性方面可能也会有问题。

4.3.2 mall-admin

本章介绍管理后台的前端模块。

`mall-admin` 模块的代码基于 `vue-element-admin`

4.3.2.1 角色页面

管理员的角色页面用于创建角色和分配权限。

其中点击授权按钮，则会出现系统所有的权限和当前已经分配的权限，具体代码可以见 `AdminRoleController` 的 `getPermissions`。系统所有的权限，可以通过后端自动解析权限描述注解来获取；当前已分配权限，可以通过数据库访问来获取。

4.3.2.2 权限 API

`vue-element-admin` 中权限校验是基于 `role` 完成的权限校验，而其他一些开源项目中权限校验是基于 `permission` 权限值完成的权限校验，而本项目进行了调整，基于 `permission URL` 完成权限校验。

后端权限校验逻辑仍然是权限值，而前端权限校验逻辑则是 URL。例如，后端开发了一个 API 是 `/admin/ad/list`，后端访问所需的权限值是 `admin:ad:list`，此时前端则不需要考虑权限值，而是仅仅需要知道自己访问 API 的方式，也就是在菜单或者页面按钮的地方写上 `GET/admin/ad/list`，而不是 `admin:ad:list`。之所以是可行的，是因为后端权限是一个（权限值，权限描述，权限访问 API）构成的 `Permission`

类。

当然，这里也存在一定的局限性。

4.3.2.3 菜单权限

见 `src/route/index.js` 代码

例如

```
{
  path: '/promotion',
  component: Layout,
  redirect: 'noredirect',
  alwaysShow: true,
  name: 'promotionManage',
  meta: {
    title: '推广管理',
    icon: 'chart'
  },
  children: [
    {
      path: 'ad',
      component: () => import('@views/promotion/ad'),
      name: 'ad',
      meta: {
        perms: ['GET /admin/ad/list', 'POST /admin/ad/create', 'GET /admin/ad/read', 'POST /admin/ad/update', 'POST /admin/ad/delete'],
        title: '广告管理',
        noCache: true
      }
    }
  ]
},
```

在菜单的 `meta` 中的 `perms` 属性声明了当前菜单所需要的操作权限。如果管理员登录以后，所分配的操作权限存在至少一个，那么该菜单就会显示；否则该菜单就会隐藏。

4.3.2.4 按钮权限

在每个页面中一些组件可以采用指令 `v-permission` 或者函数 `checkPermission`

来实现按钮权限。

例如，src/views/promotion/ad 页面中

```
<div class="filter-container">
  <el-input v-model="listQuery.name" clearable class="filter-item"
style="width: 200px;" placeholder="请输入广告标题"/>
  <el-input v-model="listQuery.content" clearable class="filter-item"
style="width: 200px;" placeholder="请输入广告内容"/>
  <el-button v-permission="['GET /admin/ad/list']" class="filter-item"
type="primary" icon="el-icon-search" @click="handleFilter">查找</el-button>
  <el-button v-permission="['POST /admin/ad/create']" class="filter-item"
type="primary" icon="el-icon-edit" @click="handleCreate">添加</el-button>
  <el-button :loading="downloadLoading" class="filter-item" type="primary"
icon="el-icon-download" @click="handleDownload">导出</el-button>
</div>
```

element 的 el-button 组件声明了操作权限是 GET /admin/ad/list'。如果管理员登录以后，所分配的操作权限匹配，那么该按钮就会显示；否则该按钮就会隐藏。

4.3.2.5 权限局限性

前端权限这里的代码调整旨在解决一些认为不合理的地方，但是实际上也同时带来了一定的局限性或者限制。

4.4 前后端接口设计

4.4.1 请求格式

这里没有采用 RESTful 风格的接口，而是定义具体语义的接口。目前只使用 GET 和 POST 来表示请求内容和更新内容两种语义。

(1) GET 请求

GET API_URL?params

例如

GET /home/index

或者

GET /goods/list?page=1&limit=10

(2) POST 更新

POST API_URL

```
{
  body
}
```

例如

```
POST /cart/clear
```

或者

```
POST /goods/star
{
  id: 1
}
```

(3) 分页请求参数

当 GET 请求后端获取数组数据时，需要传递分页参数。

例如

```
GET /goods/list?page=1&limit=10&sort=add_time&order=desc
```

本项目的通用分页请求参数统一传递四个：

page: 请求页码。和通常计算机概念中数组下标从 0 开始不同，这里的 page 参数应该从 1 开始，1 即代表第一页数据；

limit: 每一页数量

sort: 排序字段。例如 “add_time” 或者 “id”；

order: 升序降序。只能是 “desc” 或者 “asc”。

此外，这里四个参数是可选的，后端应该设置默认参数，因此即使前端不设置，后端也会自动返回合适的对象数组响应数据。

4.4.2 响应格式

```
Content-Type: application/json;charset=UTF-8
```

```
{
  body
}
```

而 body 是存在一定格式的 json 内容：

```
{
  errno: xxx,
  errmsg: xxx,
```

```
data: {}  
}
```

(1) 失败异常

```
{  
  errno: xxx,  
  errmsg: xxx  
}
```

- `errno` 是错误码，具体语义见 4.4.3 节。

- `errmsg` 是错误信息。

(2) 操作成功

```
{  
  errno: 0,  
  errmsg: "成功",  
}
```

(3) 普通对象

```
{  
  errno: 0,  
  errmsg: "成功",  
  data: {}  
}
```

(4) 数组对象

```
{  
  errno: 0,  
  errmsg: "成功",  
  data: {  
    list: [],  
    total: XX,  
    page: XX,  
    limit: XX,  
    pages: XX  
  }  
}
```

`list` 是对象数组，`total` 是总的数量。

4.4.3 错误码

（1）系统通用错误码

系统通用错误码包括 4XX 和 5XX

4xx，前端错误，说明前端开发者需要重新了解后端接口使用规范：

401，参数错误，即前端没有传递后端需要的参数；

402，参数值错误，即前端传递的参数值不符合后端接收范围。

5xx，后端系统错误，除 501 外，说明后端开发者应该继续优化代码，尽量避免返回后端系统错误码：

501，验证失败，即后端要求用户登录；

502，系统内部错误，即没有合适命名的后端内部错误；

503，业务不支持，即后端虽然定义了接口，但是还没有实现功能；

504，更新数据失效，即后端采用了乐观锁更新，而并发更新时存在数据更新失效；

505，更新数据失败，即后端数据库更新失败（正常情况应该更新成功）。

（2）商场业务错误码

- AUTH_INVALID_ACCOUNT = 700
- AUTH_CAPTCHA_UNSupport = 701
- AUTH_CAPTCHA_FREQUENCY = 702
- AUTH_CAPTCHA_UNMATCH = 703
- AUTH_NAME_REGISTERED = 704
- AUTH_MOBILE_REGISTERED = 705
- AUTH_MOBILE_UNREGISTERED = 706
- AUTH_INVALID_MOBILE = 707
- AUTH_OPENID_UNACCESS = 708
- AUTH_OPENID_BINDED = 709
- GOODS_UNSHelve = 710
- GOODS_NO_STOCK = 711
- GOODS_UNKNOWN = 712
- GOODS_INVALID = 713
- ORDER_UNKNOWN = 720
- ORDER_INVALID = 721
- ORDER_CHECKOUT_FAIL = 722

- ORDER_CANCEL_FAIL = 723
- ORDER_PAY_FAIL = 724
- ORDER_INVALID_OPERATION = 725
- ORDER_COMMENTED = 726
- ORDER_COMMENT_EXPIRED = 727
- Groupon_EXPIRED = 730
- COUPON_EXCEED_LIMIT = 740
- COUPON_RECEIVE_FAIL = 741
- COUPON_CODE_INVALID = 742

(3) 管理后台业务错误码

- ADMIN_INVALID_NAME = 601
- ADMIN_INVALID_PASSWORD = 602
- ADMIN_NAME_EXIST = 602
- ADMIN_ALTER_NOT_ALLOWED = 603
- ADMIN_DELETE_NOT_ALLOWED = 604
- ADMIN_INVALID_ACCOUNT = 605
- GOODS_UPDATE_NOT_ALLOWED = 610
- GOODS_NAME_EXIST = 611
- ORDER_CONFIRM_NOT_ALLOWED = 620
- ORDER_REFUND_FAILED = 621
- ORDER_REPLY_EXIST = 622
- USER_INVALID_NAME = 630
- USER_INVALID_PASSWORD = 631
- USER_INVALID_MOBILE = 632
- USER_NAME_EXIST = 633
- USER_MOBILE_EXIST = 634
- ROLE_NAME_EXIST = 640
- ROLE_SUPER_SUPERMISSION = 641
- ROLE_USER_EXIST = 642

4.4.4 Token

前后端采用 token 来验证访问权限。

```
POST /wx/auth/login
{
  "username": "user123",
  "password": "user123"
}
```

```
{
  "errno": 0,
  "data": {
    "userInfo": {
      "nickName": "user123",
      "avatarUrl": "https://wpimg.wallstcn.com/f778738c-e4f8-4870-b634-56703b4acafe.gif"
    },
    "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0aGlzIGlzIGxpdGVtYWxsIH  
Rva2VuIiwiaXNjaXkiOiJ0aXNjaXkiLCJpdiI6IjEwMDAwMDAwMDAwMDAwMDAwMDA  
MzI2ODUwLCJ1c2VySWQiOiJESlmlhdCI6MTU1NzMxOTY1MH0.XP0TuhupV_ttQs  
Cr1KTaPZVITbVzVOcnq_K0kXdabri0"
    },
    "errmsg": "成功"
  }
}
```

GET http://localhost:8080/wx/address/list
X-Mall-Token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0aGlzIGlzIGxpZGVtYWxsIH

```
Rva2VuIwiYXVkJjoiTUlOSUFQUCIsImlzcyI6IkxJVEVNQUxMIwiZXhwIjoxNTU3MzM2ODU0LCJ1c2VySWQiOjIsImlhdCI6MTU1NzMyOTY1NH0.JY1-cqOnmi-CVjFohZMqK2iAdAH4O6CKj0Cqd5tMF3M
```

3. 管理后台自定义 Header

访问受保护管理后台 API 则是自定义 X-Mall-Admin-Token 头部。

(1) 管理后台前端访问管理后台后端登录 API/admin/auth/login

```
POST /admin/auth/login

{
  "username": "admin123",
  "password": "admin123"
}
```

(2) 成功以后，管理后台前端会接收后端响应的一个 token，

```
{
  "errno": 0,
  "data": {
    "adminInfo": {
      "nickName": "admin123",
      "avatar": "https://wpimg.wallstcn.com/f778738c-e4f8-4870-b634-56703b4acafe.gif"
    },
    "token": "f2dbcae8-6e25-4f8e-bc58-aa81d512c952"
  },
  "errmsg": "成功"
}
```

(3) 请求受保护 API 时，则采用自定义头部携带此 token

```
GET http://localhost:8080/wx/address/list
X-Mall-Admin-Token: f2dbcae8-6e25-4f8e-bc58-aa81d512c952
```

4.4.5 API 服务列表

本系统使用的部分前后端 API 如表 4-1 所示。

表 4-1 本系统使用的部分 API 统计表

应用场景	接口链接	请求参数
商品列表	GET /wx/goods/list	isNew: 是否新品, true 或

		<p>者 false</p> <p>isHot: 是否热卖商品, true 或者 false</p> <p>keyword: 关键字, 如果设置则查询是否匹配关键字</p> <p>brandId: 品牌商 ID, 如果设置则查询品牌商所属商品</p> <p>categoryId: 商品分类 ID, 如果设置则查询分类所属商品</p> <p>page: 请求页码</p> <p>limit: 每一页数量</p> <p>sort: 排序字段</p> <p>order: 升序降序</p>
商品详情	GET /wx/goods/detail	<p>id: 商品 ID, 例如 id=1152008</p>
商品推荐	GET /wx/goods/related	<p>id: 商品 ID, 例如 id=1152008</p> <p>page: 请求页码</p> <p>limit: 每一页数量</p> <p>sort: 排序字段</p> <p>order: 升序降序</p>
在售商品总数	GET /wx/goods/count	无
订单列表	GET /wx/order/list	<p>showType: 订单类型, 0 则全部, 1 则待付款, 2 则待发货, 3 则待收货, 4 则代评价</p> <p>page: 请求页码</p> <p>limit: 每一页数量</p> <p>sort: 排序字段</p> <p>order: 升序降序</p>
订单详情	GET /wx/order/detail	orderId : 订单 ID
订单删除	POST /wx/order/delete	orderId : 订单 ID
篇幅所限, 全部 API 请参阅代码		

5 系统使用与测试

5.1 用户管理模块测试

(1) 会员管理：查找指定名称或手机号的会员，显示用户的详细信息，对用户信息进行导出等；

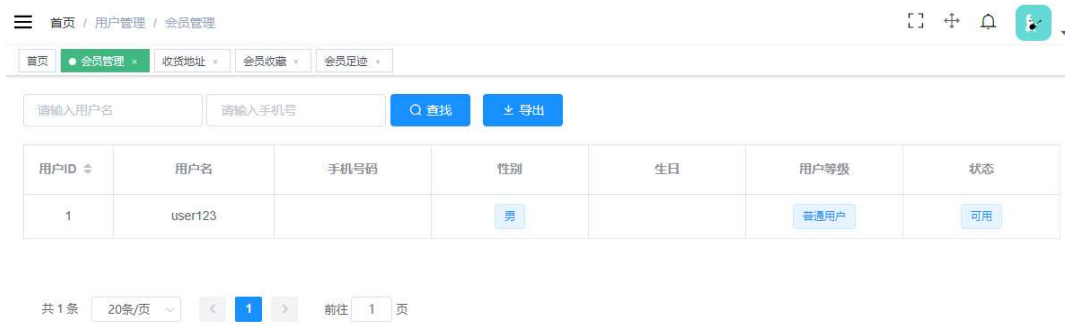


图 5-1 会员管理

(2) 收货地址：查找指定 ID 或收货人名称的地址，显示地址的详细信息，对地址信息进行导出等；



图 5-2 收货地址

(3) 会员收藏：查找指定用户或指定商品的收藏信息，显示收藏的详细信息，对收藏信息进行导出等；

三 首页 / 用户管理 / 会员收藏

首页 会员管理 收货地址 会员收藏 会员足迹

请输入用户ID 请输入商品ID 查找 导出

收藏ID	用户ID	商品ID	添加时间
38	1	1116011	2020-06-19 10:34:08
35	1	1152161	2020-06-18 15:12:02
33	1	1075023	2020-06-18 13:54:09
30	1	1181036	2020-06-18 09:22:49
29	1	1084001	2020-06-17 14:20:15
28	1	1030006	2020-06-15 16:46:41

共 6 条 20条/页 < 1 > 前往 1 页

图 5-3 会员收藏

（4）会员足迹：查找指定用户或指定商品的浏览记录，显示浏览记录的详细信息，对浏览信息进行导出等；

三 首页 / 用户管理 / 会员足迹

首页 会员管理 收货地址 会员收藏 会员足迹

请输入用户ID 请输入商品ID 查找 导出

足迹ID	用户ID	商品ID	添加时间
626	1	1181037	2020-06-19 13:18:37
625	1	1084003	2020-06-19 12:54:19
624	1	1019002	2020-06-19 11:51:26
623	1	1181042	2020-06-19 11:42:44
622	1	1037011	2020-06-19 10:37:09
621	1	1116011	2020-06-19 10:33:59
620	1	1116011	2020-06-19 10:28:09
619	1	1116011	2020-06-19 10:27:23

图 5-4 会员足迹

（5）搜索历史：查找指定用户或关键字的搜索记录，显示搜索记录的详细信息，对搜索信息进行导出等；

三 首页 / 用户管理 / 搜索历史

首页

会员管理

收货地址

会员收藏

会员足迹

搜索历史

请输入用户ID

请输入搜索历史关键字

查找

导出

搜索ID	用户ID	关键字	添加时间
56	1	s	2020-06-19 10:10:23
55	1	a	2020-06-19 10:10:08
54	1	哈哈	2020-06-19 09:52:07
53	1	001	2020-06-19 09:28:53
52	1	垃圾桶	2020-06-18 15:52:27
51	1	1	2020-06-18 14:07:06
50	1	11	2020-06-18 14:07:01
49	1	毛	2020-06-18 11:47:26
48	1	毛	2020-06-18 11:47:24
47	1	请	2020-06-18 11:47:19
46	1	520元礼包抢先领	2020-06-18 11:01:59
45	1	测试	2020-06-18 11:01:27

图 5-5 搜索历史

（6）意见反馈：查找指定用户或反馈 ID 的反馈记录，显示反馈记录的详细信息，对反馈信息进行导出等；

三 首页 / 用户管理 / 意见反馈

首页

会员管理

收货地址

会员收藏

会员足迹

搜索历史

意见反馈

请输入用户名

请输入反馈ID

查找

导出

反馈ID	用户名	手机号码	反馈类型	反馈内容	反馈图片	时间
2	user123	13376542345	功能异常	喝好		2020-06-15 03:00:42
1	user123	13363342561	其他	很好,反馈个异常问题		2020-06-12 21:22:01

共 2 条

20条/页

< 1 >

前往 1 页

图 5-6 意见反馈

5.2 商场&商品管理模块测试

（1）商场信息管理：发货行政区域的查询功能，品牌制造商和商品类目的增删改查功能；

三 首页 / 商场管理 / 商品类目

商品类目

添加

类目ID	类目名	类目图标	类目图片	关键字	简介	级别	操作
> 1036026	化妆品			化妆品		一级类目	编辑 删除
> 1036030	衣服				衣服类	一级类目	编辑 删除
> 1036033	洋酒			洋酒		一级类目	编辑 删除
1036036	dfgdfgfdg					二级类目	编辑 删除
1036037	dfgdfgfdg					二级类目	编辑 删除
1036038	车					一级类目	编辑 删除

图 5-7 商场信息管理

(2) 订单管理：查找指定用户或订单 ID 或订单状态的订单记录，显示订单记录的详细信息，对订单信息进行导出，打印订单信息等；

三 首页 / 商场管理 / 订单管理

订单管理

请输入用户ID 请输入订单编号 开始日期 至 结束日期 请选择订单状态 查找 导出

订单编号	用户ID	订单状态	订单金额	支付金额	支付时间	物流单号	物流渠道	操作
20200619453524	1	系统取消	44	44				详情 删除
20200619882798	1	系统取消	199	199				详情 删除
20200619994051	1	系统取消	37	37				详情 删除
20200619365931	1	用户取消	94	94				详情 删除
20200619251536	1	系统取消	89	89				详情 删除
20200619507658	1	系统取消	198	198				详情 删除
20200618892451	1	系统取消	67	67				详情 删除
20200618733582	1	系统取消	37	37				详情 删除
20200618972033	1	用户取消	44	44				详情 删除
20200618532087	1	用户取消	5962	5962				详情 删除

图 5-8 订单管理

(3) 通用问题：设置通用问题内容的快捷回复，查询指定关键词的通用问题，对通用问题进行增删改处理，对通用问题进行导出等；



图 5-9 通用问题

(4) 关键词: 设置关键词对应的推广链接, 开启或关闭推荐状态, 查询指定关键词或跳转链接对应的推广内容, 对推广内容进行增删改处理, 对关键词和跳转链接进行导出等;

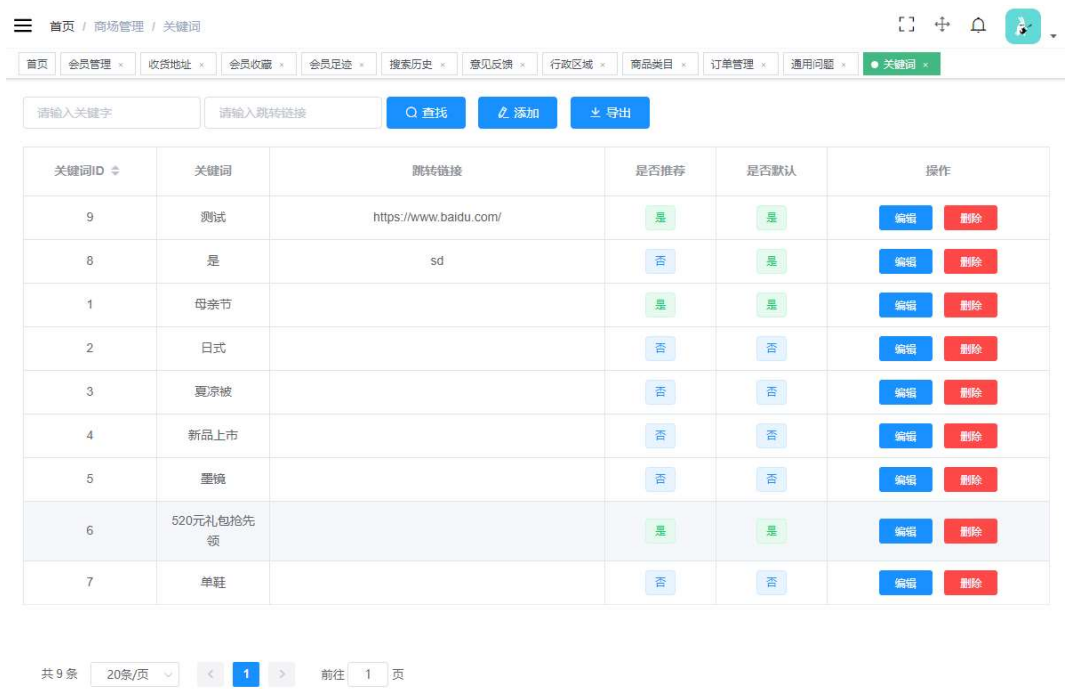


图 5-10 关键词

(5) 商品列表: 查找指定编号或名称对应的商品信息, 显示商品记录的名称、图片、价格、在售、规格、库存等一系列详细信息, 对商品列表进行导出、更新和删除等;

三 首页 / 商品管理 / 商品列表

首页 会员管理 收货地址 会员收藏 会员足迹 搜索历史 意见反馈 行政区域 商品类目 订单管理 通用问题 关键词 商品列表

请输入商品ID

请输入商品编号

请输入商品名称

Q 查找

添加

导出







	商品ID	名称	图片	分享图	详情	市场售价	当前价格	是否新品	是否热品	是否在售	操作
>	1181044	小童上衣			查看	200	0	新品	非热品	在售	编辑 删除
>	1181042	女童上衣			查看	100	200	新品	非热品	在售	编辑 删除
>	1181041	阿萨德			查看	12	10	新品	非热品	未售	编辑 删除
>	1181040	反反复复			查看	0	0	新品	非热品	未售	编辑 删除
>	1181039	哈哈			查看	9999	66	新品	非热品	在售	编辑 删除
>	1181038	12			查看	12	0	非新品	非热品	在售	编辑 删除
>	1181037	对对对			查看	3	0	新品	非热品	在售	编辑 删除
>	1138000	日式蓬软大鼓抱枕			查看	49	29	新品	非热品	在售	编辑 删除
>	1075024	升级款纯棉静音白鹅羽绒被			查看	2419	2399	非新品	非热品	在售	编辑 删除
>	1010000	澳洲纯羊毛盖毯 加厚款			查看	419	399	非新品	非热品	在售	编辑 删除
>	1138001	北欧简约山形纹绣花抱枕			查看	99	79	非新品	非热品	在售	编辑 删除
>	1074001	男式莫代尔无痕内裤			查看	79	59	非新品	非热品	在售	编辑 删除
>	1010001	澳洲纯羊毛盖毯 舒适款			查看	319	299	非新品	非热品	在售	编辑 删除

图 5-11 商品列表

（6）商品上架：新增商品的信息；

三

首页 / 商品管理 / 商品上架

商品上架

商品上架

商品介绍

商品编号

商品名称

市场价0.00元

是否新品

新品

非新品

是否热卖

普通

热卖

是否在售

在售

未售

商品图片

宣传画廊

商品单位

件 / 个 / 盒

关键字

+ 增加

所属分类

请选择

所属品牌商

请选择

商品简介

图 5-12 商品上架

（7）商品评论：查找指定用户或商品对应的商品评论，显示商品评论的打分和评论内容等信息，对商品评论进行回复或删除操作；

三

首页 / 商品管理 / 商品评论

商品评论

商品评论

请输入用户ID

请输入商品ID

查找

导出




用户ID	商品ID	打分	评论内容	评论图片	时间	操作
1	1009024	1	很舒服的懒人沙发，放在客厅小阳台，忍不住幻想回家后坐着读一本书喝一杯咖啡晒一晒太阳的幸福生活啊！		2018-02-01 00:00:00	<div>回复删除</div>
1	1020000	1	一直想买个腰靠，终于在严选下手了，大小合适，软硬也适中，包装袋尤其惊喜，软软的，还可以拿来装装别的东西		2018-02-01 00:00:00	<div>回复删除</div>
1	1027004	1	特价入的，面料手感挺好，夏天到了正好可以用了。		2018-02-01 00:00:00	<div>回复删除</div>

图 5-13 商品评论

5.3 系统管理模块测试

(1) 管理员：对管理员进行增删改查操作，为管理员添加或删除不同角色，导出管理员记录等；

三 首页 / 系统管理 / 管理员

首页 商品上架 商品评论 管理员

请输入管理员名称 查找 添加 导出

管理员ID	管理员名称	管理员头像	管理员角色	操作
22	sisterMen		推广管理员	编辑 删除
17	crradvs		超级管理员 推广管理员 商场管理员	编辑 删除
15	aaaaaa		推广管理员 商场管理员	编辑 删除
4	promotion123		推广管理员	编辑 删除
1	admin123		超级管理员	编辑 删除

共 5 条 20条/页 1 前往 1 页

图 5-14 管理员

(2) 操作日志：显示后台管理系统的一系列操作行为日志，包括操作角色、IP 地址、时间、相关操作等均被记录在日志中，并提供操作日志的查找筛选；

三 首页 / 系统管理 / 操作日志

首页 商品上架 商品评论 管理员 操作日志

请输入操作管理员 查找

操作管理员	IP地址	操作时间	操作类别	操作动作	操作状态	操作结果	备注信息
admin123	220.248.70.98	2020-06-19 13:34:47	安全操作	登录	成功		
admin123	121.56.100.27	2020-06-19 13:25:08	安全操作	登录	成功		
admin123	42.91.160.154	2020-06-19 13:24:39	安全操作	登录	成功		
admin123	183.195.155.118	2020-06-19 13:23:35	安全操作	登录	成功		
admin123	183.195.12.4	2020-06-19 13:16:40	安全操作	登录	成功		
admin123	114.244.75.46	2020-06-19 13:09:43	安全操作	登录	成功		

图 5-15 操作日志

(3) 角色管理：对角色进行增删改查操作，为角色授予或取消授予不同的后台管理权限；



图 5-16 角色管理

(4) 对象存储：按照对象 KEY 或对象名称进行对象数据的查找，显示对象存储的详细信息，对对象存储数据进行增删改操作，将对象进行导出；

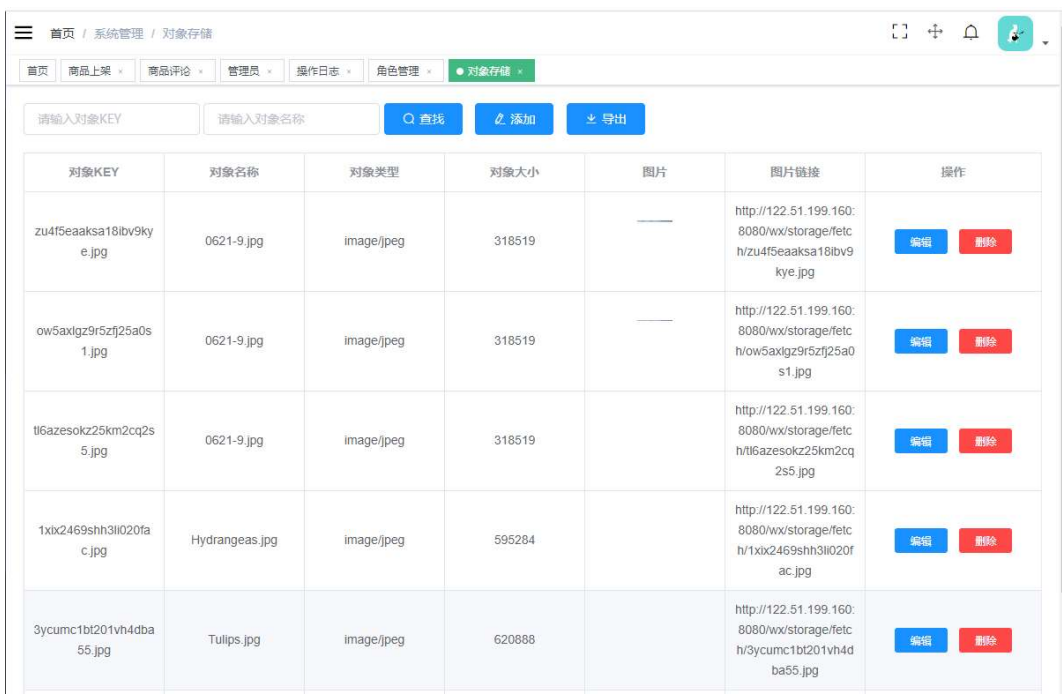


图 5-17 对象存储

5.4 配置管理模块测试

(1) 商场配置：对后台管理系统对应电商站点的商场名称、商场地址、联系电话等信息进行编辑操作；

三

首页 / 配置管理 / 商场配置

全屏

刷新

消息

头像

首页

商场配置

* 商场名称

非洲商

* 商场地址

非洲

地理坐标

121.587839

-

31.201900

* 联系电话

010-xx-xx

* 联系QQ

123456

取消

确定

图 5-18 商场配置

(2) 运费配置：配置订单所需运费的相关规则；

三

首页 / 配置管理 / 运费配置

全屏

刷新

消息

头像

首页

商场配置

运费配置

* 运费满减所需最低消费

88

* 运费满减不足所需运费

8

取消

确定

图 5-19 运费配置

(3) 订单配置：配置订单中的下单超时、发货超期、自动确认收货等相关规则；

三

首页 / 配置管理 / 订单配置

全屏

刷新

消息

头像

首页

商场配置

运费配置

订单配置

* 用户下单后超时

60

分钟

用户未付款，则订单自动取消

* 订单发货后超期

7

天

未确认收货，则订单自动确认收货

* 确认收货后超期

7

天

未评价商品，则取消评价资格

取消

确定

图 5-20 订单配置

5.5 统计报表模块测试

(1) 用户统计：对用户增长情况进行统计分析；



图 5-21 用户统计

(2) 订单统计：对订单按照数量、用户、价格等方面进行统计分析；



图 5-22 订单统计

(3) 商品统计：对商品订单量、下单货物数量、下单货品总额进行统计分析；



图 5-23 商品统计