# Shopping-Ecommerce 电商平台项目说明文档

版本：v1.0.0  
日期：2024年12月  
作者：项目团队

# 目录

1. 项目概述  
2. 系统架构  
3. 技术选型  
4. 功能模块详述  
5. 数据库设计  
6. API接口文档  
7. 前端项目说明  
8. 后端项目说明  
9. Docker容器化部署  
10. 开发进度与规划

## 1. 项目概述

Shopping-Ecommerce 是一个基于现代技术栈开发的全栈电商平台，采用前后端分离架构设计。项目旨在提供完整的电商解决方案，包括商家管理、商品管理、订单处理、用户管理、支付系统等核心功能模块。  
  
项目特点：  
• 微服务架构：后端采用Spring Boot微服务架构，模块化设计，便于扩展和维护  
• 现代前端：基于Vue 3 + TypeScript + Element Plus构建，提供良好的用户体验  
• 高性能：集成Redis缓存，优化数据库查询性能  
• 安全可靠：集成Spring Security，提供完善的认证授权机制  
• 容器化部署：支持Docker容器化部署，简化运维操作  
• 完整文档：提供详细的API文档和部署指南

## 2. 系统架构

系统采用前后端分离的微服务架构，主要分为以下几个层次：  
  
表示层（Presentation Layer）：  
• 前端应用：基于Vue 3的单页面应用，提供用户界面  
• 移动端：预留接口，支持移动端应用开发  
  
业务层（Business Layer）：  
• API网关：统一入口，处理路由和负载均衡  
• 微服务集群：按业务功能划分的独立服务  
• 认证服务：统一的用户认证和授权  
  
数据层（Data Layer）：  
• MySQL：主数据库，存储业务数据  
• Redis：缓存层，提升系统性能  
• 文件存储：处理图片、文档等文件上传  
  
基础设施层（Infrastructure Layer）：  
• Docker容器：服务容器化部署  
• 监控系统：日志收集和性能监控  
• 配置中心：统一配置管理

## 3. 技术选型

### 3.1 前端技术栈

|  |  |  |
| --- | --- | --- |
| 技术 | 版本 | 作用 |
| Vue.js | 3.x | 渐进式前端框架 |
| TypeScript | 5.x | 类型安全的JavaScript |
| Vite | 5.x | 快速构建工具 |
| Element Plus | 2.x | Vue3 UI组件库 |
| Pinia | 2.x | 状态管理 |
| Vue Router | 4.x | 路由管理 |

### 3.2 后端技术栈

|  |  |  |
| --- | --- | --- |
| 技术 | 版本 | 作用 |
| Spring Boot | 3.4.5 | 微服务框架 |
| Spring Security | 6.x | 安全认证框架 |
| MySQL | 8.0 | 关系型数据库 |
| Redis | 7.x | 内存缓存数据库 |
| MyBatis | 3.x | ORM数据映射框架 |
| Maven | 3.x | 项目构建工具 |
| Docker | Latest | 容器化部署 |

## 4. 功能模块详述

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 功能描述 | 开发进度 | 备注 |
| 商家模块 | 商家注册、审核、信息管理 | 95% | 已完成基础功能 |
| 文件模块 | 文件上传、存储、访问管理 | 90% | 支持多种文件格式 |
| 用户模块 | 用户注册、登录、个人信息管理 | 40% | 在开发中 |
| 商品目录 | 商品分类、信息管理、搜索 | 35% | 基础架构已搭建 |
| 库存模块 | 库存管理、出入库记录 | 10% | 待开发 |
| 购物车 | 购物车管理、商品加车、结算 | 5% | 待开发 |
| 订单模块 | 订单创建、支付、状态管理 | 25% | 基础数据结构已完成 |
| 物流模块 | 物流跟踪、配送管理 | 8% | 待开发 |
| 支付模块 | 多种支付方式集成 | 5% | 待开发 |
| 认证授权 | 用户认证、权限管理 | 15% | 基础框架已搭建 |

## 5. 数据库设计

系统采用MySQL 8.0作为主数据库，使用Flyway进行数据库版本管理。主要数据表结构如下：

### 5.1 商家相关表

|  |  |  |  |
| --- | --- | --- | --- |
| 表名 | 字段名 | 类型 | 说明 |
| merchants | id | BIGINT | 主键 |
|  | merchant\_name | VARCHAR(255) | 商家名称 |
|  | contact\_person | VARCHAR(100) | 联系人 |
|  | contact\_phone | VARCHAR(20) | 联系电话 |
|  | email | VARCHAR(255) | 邮箱地址 |
|  | address | TEXT | 地址 |
|  | status | VARCHAR(20) | 状态 |
|  | created\_at | TIMESTAMP | 创建时间 |
|  | updated\_at | TIMESTAMP | 更新时间 |

## 6. Docker容器化部署

项目支持完整的Docker容器化部署，通过docker-compose一键启动所有服务。容器架构包括：  
  
• MySQL数据库容器（端口3306）  
• Redis缓存容器（端口6379）  
• 商家服务容器（端口8081）  
• 文件服务容器（端口8082）  
• 前端应用容器（端口3000）  
  
部署优势：  
• 一键启动：通过start-docker.bat脚本一键部署  
• 环境隔离：每个服务运行在独立容器中  
• 数据持久化：数据库和文件存储支持数据卷持久化  
• 易于扩展：支持横向扩展和负载均衡  
• 版本管理：支持容器镜像版本管理

### 6.1 容器服务列表

|  |  |  |  |
| --- | --- | --- | --- |
| 服务名称 | 端口 | 说明 | 数据持久化 |
| shopping-frontend | 3000 | Vue.js 前端应用 | 无 |
| shopping-merchant-service | 8081 | 商家管理服务 | 无 |
| shopping-file-service | 8082 | 文件上传服务 | file\_uploads |
| shopping-mysql | 3306 | MySQL 数据库 | mysql\_data |
| shopping-redis | 6379 | Redis 缓存 | redis\_data |

### 6.2 快速部署指南

前置要求：  
1. 安装Docker Desktop for Windows  
2. 确保Docker服务正在运行  
3. 系统要求：Windows 10/11，至少8GB内存  
  
部署步骤：  
1. 双击运行start-docker.bat脚本  
2. 等待所有容器启动完成（约2-3分钟）  
3. 访问http://localhost:3000查看前端应用  
  
访问地址：  
• 前端应用：http://localhost:3000  
• 商家管理API：http://localhost:8081/merchant  
• 文件服务：http://localhost:8082  
• MySQL数据库：localhost:3306（用户：root，密码：123456）  
• Redis缓存：localhost:6379  
  
停止服务：  
双击运行stop-docker.bat脚本即可停止所有容器服务。

## 7. 开发进度与规划

### 7.1 当前进度

已完成模块（85-100%）：  
• 商家模块：商家注册、审核、信息管理功能完整  
• 文件模块：支持多种文件格式上传和管理  
• 通用模块：基础工具类和配置  
• 前端商家界面：商家管理相关页面  
  
开发中模块（20-40%）：  
• 用户模块：基础用户管理功能  
• 商品目录：商品分类和基础管理  
• 订单模块：订单数据结构设计  
  
待开发模块（5-10%）：  
• 库存管理、购物车、物流跟踪  
• 支付集成、认证授权系统

### 7.2 下一步计划

短期目标（1-2个月）：  
1. 完善用户模块功能  
2. 完成商品目录核心功能  
3. 实现基础购物车功能  
4. 集成认证授权系统  
  
中期目标（3-6个月）：  
1. 完成订单处理流程  
2. 集成支付系统  
3. 实现库存管理  
4. 开发物流跟踪功能  
  
长期目标（6个月以上）：  
1. 性能优化和压力测试  
2. 移动端应用开发  
3. 数据分析和报表系统  
4. 第三方服务集成扩展

## 8. 总结

Shopping-Ecommerce电商平台是一个基于现代技术栈的全栈项目，具有以下特点：  
  
技术优势：  
• 采用微服务架构，模块化设计，便于维护和扩展  
• 前后端分离，技术栈先进，开发效率高  
• 支持Docker容器化部署，运维简单  
• 完善的文档和代码规范  
  
项目亮点：  
• 完整的电商业务流程设计  
• 高性能的缓存和数据库优化  
• 安全可靠的认证授权机制  
• 良好的用户体验设计  
  
发展前景：  
项目具有良好的扩展性和可维护性，可以根据业务需求持续迭代和优化。通过模块化的设计，可以灵活地添加新功能和集成第三方服务，为未来的业务发展提供坚实的技术基础。  
  
本文档将随着项目的发展持续更新，确保文档与代码的同步性。