

我的Github开源项目，从0到20000 Star!

原创 梦想de星空 macrozheng 2019-08-21 08:31

收录于合集
#开源之路

20个

最近，我在Github上面开源的项目mall已经突破了20000 Star，这个项目是2018年3月份开始开发的，耗时9个月，发布了第一个版本，一直维护至今。回想起来，还是有诸多感慨的，下面我就谈谈我的项目发展的整个历程。

macrozheng / mall

Unwatch

1,111

Star

20,001

Fork

7,907

Code

Issues 16

Pull requests 1

Projects 0

Wiki

Security

Insights

Settings

mall项目是一套电商系统，包括前台商城系统及后台管理系统，基于SpringBoot+MyBatis实现。前台商城系统包含首页门户、商品推荐、商品搜索、商品展示、购物车、订单流程、会员中心、客户服务、帮助中心等模块。后台管理系统包含商品管理、订单管理、会员管理、促销管理、运营管理、内容管理、统计报表、财务管理、权限管理、设置等模块。<http://39.98.190.128/index.html>

spring-boot

spring-security

mybatis

docker

swagger-ui

mysql

redis

elasticsearch

mongodb

elk

spring

rabbitmq

spring-cloud

java

springcloud

springboot

Manage topics

287 commits

1 branch

1 release

1 contributor

Apache-2.0

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

macrozheng fix swagger icon bug

Latest commit c42d57c 2 days ago

document	add pos file	17 days ago
mall-admin	fix swagger icon bug	2 days ago
mall-common	部分工具类实现改用hutool工具包	4 months ago
mall-demo	项目结构改造	4 months ago
mall-mbg	修复mybatis generator重复生成mapper.xml内容的问题	2 months ago
mall-portal	fix swagger icon bug	2 days ago
mall-search	pom插件配置修改	2 months ago
.gitignore	初始项目信息导入	last year
LICENSE	Update LICENSE	4 months ago
README-DEV.md	说明文件替换	9 months ago
README.md	Update README.md	9 days ago
keystore.p12	添加https支持	
pom.xml	项目结构改造	4 months ago

macrozheng

项目发展历程

为什么要写这个项目

2018年3月的时候，我在Github上面闲逛，想要找一个业务和技术相结合的项目，但是发现很多项目都是以技术为主，业务都比较简单。于是我产生了自己写一个业务与技术相结合的项目的想法，业务选择了电商，因为这是一个大家比较容易理解的业务场景，同时有很多成熟的系统可以借鉴。技术选择了SpringBoot全家桶，因为当时SpringBoot比较火，同时自己也想学习并实践下。

明确项目需求

划分项目模块

当时有了解到一个最小精益产品的概念，就是把一个复杂的产品进行简化，简化到一个只保留核心功能的产品。使用这种方法，我对一些成熟的系统功能进行简化，最后确定了管理后台需要开发的功能为商品管理、订单管理、运营管理、促销管理、内容管理、会员管理等功能，移动端需要开发的功能为首页推荐、首页内容、我的、购物车、商品展示、订单等功能。

项 目 业 务 架 构 图 :



使用工具整理需求

当时整理需求用到了一个叫**MindMaster**思维导图工具，首先划分功能模块，之后划分每个模块中的功能，最后对每个功能所要处理的数据字段进行标注。形成了一套明确需求的思维导图，有了它，之后的数据库设计就容易多了！

当时设计的思维导图可以查看这里：[mall数据库表结构概览](#)

数据库表设计

有了上面整理需求的思维导图以后，就可以开始设计数据库了。刚开始设计数据库的时候，并不需要把数据库设计的特别完善，因为等到你编码用到时，总是要改的，只需要满足当前功能的数据存储需求即可。说说数据库的外键，数据库表之间建议做逻辑关联，不要设置外键。比如说我的项目里面的商品表，和十几张表都是有关联的，要是我用外键的话，当商品表被锁死了，其他外键关联的表也会被锁死，这样小半个数据库都会被锁死。再说说刚开始设计的时候是否需要添加索引，个人建议暂时不要加，等编码的时候再加。个人推荐数据库表使用Power Designer等设计工具来设计，效率高，可以保留表与表之间的依赖关系。

当时设计的数据库可以查看这里：[mall数据库表结构概览](#)

后端功能开发

技术选型

技术	说明	官网
Spring Boot	容器+MVC框架	https://spring.io/projects/spring-boot
Spring Security	认证和授权框架	https://spring.io/projects/spring-security
MyBatis	ORM框架	http://www.mybatis.org/mybatis-3/zh/index.html
MyBatisGenerator	数据层代码生成	http://www.mybatis.org/generator/index.html
PageHelper	MyBatis物理分页插件	http://git.oschina.net/free/Mybatis_PageHelper
Swagger-UI	文档生产工具	https://github.com/swagger-api/swagger-ui
Hibernate-Validator	验证框架	http://hibernate.org/validator/
Elasticsearch	搜索引擎	https://github.com/elastic/elasticsearch
RabbitMq	消息队列	https://www.rabbitmq.com/
Redis	分布式缓存	https://redis.io/
MongoDb	NoSql数据库	https://www.mongodb.com/
Docker	应用容器引擎	https://www.docker.com/
Druid	数据库连接池	https://github.com/alibaba/druid
OSS	对象存储	https://github.com/aliyun/aliyun-oss-java-sdk

技术	说明	官网
JWT	JWT登录支持	https://github.com/jwtkt/jjwt
LogStash	日志收集	https://github.com/logstash/logstash-logback-encoder
Lombok	简化对象封装工具	https://github.com/rzwitserloot/lombok

个人对学习后端技术的心得

当然这一堆技术，我也不是刚开发这个项目的时候就会的，有很多都是开发过程中学会的，当时也看了很多资料，我看过的资料如下：[mall学习所需知识点（推荐资料）](#)。如果是系统学习某个技术，我推荐看书，因为书的知识体系是比较全面的，而且里面几乎没有啥错误。但是看书也有个缺点，有些书里面某些技术版本比较老旧，不过不用太过担心，因为一个流行的技术的核心不会因为版本的迭代发生太大的变化，老版本的使用方式到了新版本，绝大多数都依旧适用。

前端功能开发

技术选型

技术	说明	官网
Vue	前端框架	https://vuejs.org/
Vue-router	路由框架	https://router.vuejs.org/
Vuex	全局状态管理框架	https://vuex.vuejs.org/
Element	前端UI框架	https://element.eleme.io/
Axios	前端HTTP框架	https://github.com/axios/axios
v-charts	基于Echarts的图表框架	https://v-charts.js.org/
Js-cookie	cookie管理工具	https://github.com/js-cookie/js-cookie
nprogress	进度条控件	https://github.com/rstacruz/nprogress

个人对学习前端技术的心得

说起前端技术，很多后端开发都不怎么擅长。其实前端技术学习并不是那么难，因为现在的前端技术已经发展的很成熟了，比如Vue、React、Angular都是比较成熟的前端技术。当你前端写多了之后，你就会发现写前端也无非是使用使用框架，用js写写前端逻辑，和后端的写逻辑没啥大的区别。下面我来说说我是怎么学习前端的吧，首先我确定了我要学习的是Vue，大概花了一周看了一遍Vue的官方文档，毕竟是国人开源的框架，文档对国人还是很友好的。之后选择了一个脚手架vue-element-admin，然后大概看了一遍里面使用的技术，对这些技术都到官方网站上面看了一遍文档，主要还是看到Element的文档。之后我就拿着这个脚手架开始写我的项目实战了。学习编程，光靠看效果并不好，还是要多实践，学以致用才行！

设计移动端原型

为什么要设计一个移动端原型呢？主要是为了整个项目有个完整的业务流程，同时为下阶段移动端的开发做准备。目前我的原型有完整的移动端流程，可以完美对接后台管理。我觉得开发也需要有一定的产品设计能力，举个例子：要是某天老板有个演示产品要做，叫你去做怎么办？你要是会设计产品原型，就只要用工具做个就好了，就不用写一些临时的代码了，比开发个演示产品要省时省力的多。

移动端演示地址：<http://39.98.190.128/mall-app/mainpage.html>

项目部署

项目起初只有一套开发环境的windows部署方案，后来加入了linux部署方案，采用的docker容器化部署，之后又加入了更方便的docker-compose部署方案。

具体方案如下：

- [mall在Windows环境下的部署](#)
- [mall在Linux环境下的部署（基于Docker容器）](#)
- [mall在Linux环境下的部署（基于Docker Compose）](#)

一个完整的流程

其实做这个项目，对我来说也是一个完善自身技术栈的过程。通过这个项目，我学习到了产品、开发、运维的一系列技术，虽然不精，但都是实用的。

项目框架升级

在2019年3月的时候，进行了一次框架升级，将SpringBoot从1.5.14版本升级到了2.1.3，同时将Elasticsearch从2.3.6版本升级到了6.2.2。

完善项目文档

我觉得一个好的项目，需要一份完善的项目文档，以便更多的人来学习，于是2019年5月的时候我开始完善整个项目的文档，对整个项目的架构、业务、技术要点进行全方位的解析。

项目文档地址：**<https://github.com/macrozheng/mall-learning>**





抽取项目骨架

为了方便只需要使用mall项目的技术栈来开发自己业务系统的朋友，我将mall项目中使用的技术栈抽取出来做成了一个项目骨架，对其中的业务进行精简，只保留了核心的12张表，方便开发使用，可以自由定制业务逻辑。

项目地址：<https://github.com/macrozheng/mall-tiny>

项目Star增长历程

我的项目是从2018年12月，陆续有Star增长的，其实你只要用心去写一个开源项目，总是会有人来关注的，附上一张mall项目的Star增长图。



项目地址

mall项目地址: <https://github.com/macrozheng/mall>

推荐阅读

- [Postman: API接口调试利器](#)
- [营销模块数据库表解析: 首页内容推荐功能](#)
- [营销模块数据库表解析: 优惠券功能](#)
- [营销模块数据库表解析: 限时购功能](#)
- [mall-tiny: 基于SpringBoot+MyBatis的单应用项目骨架](#)
- [漫话: 如何给女朋友解释鸿蒙OS是怎样实现跨平台的?](#)
- [这次要是讲不明白Spring Cloud核心组件, 那我就白编这故事了](#)



欢迎关注，点个在看

阅读原文

喜欢此内容的人还喜欢

项目中到底该不该用Lombok?

macrozheng

