



链滴

# springboot+ 中间件实现秒杀系统感悟

作者: [sirwsl](#)

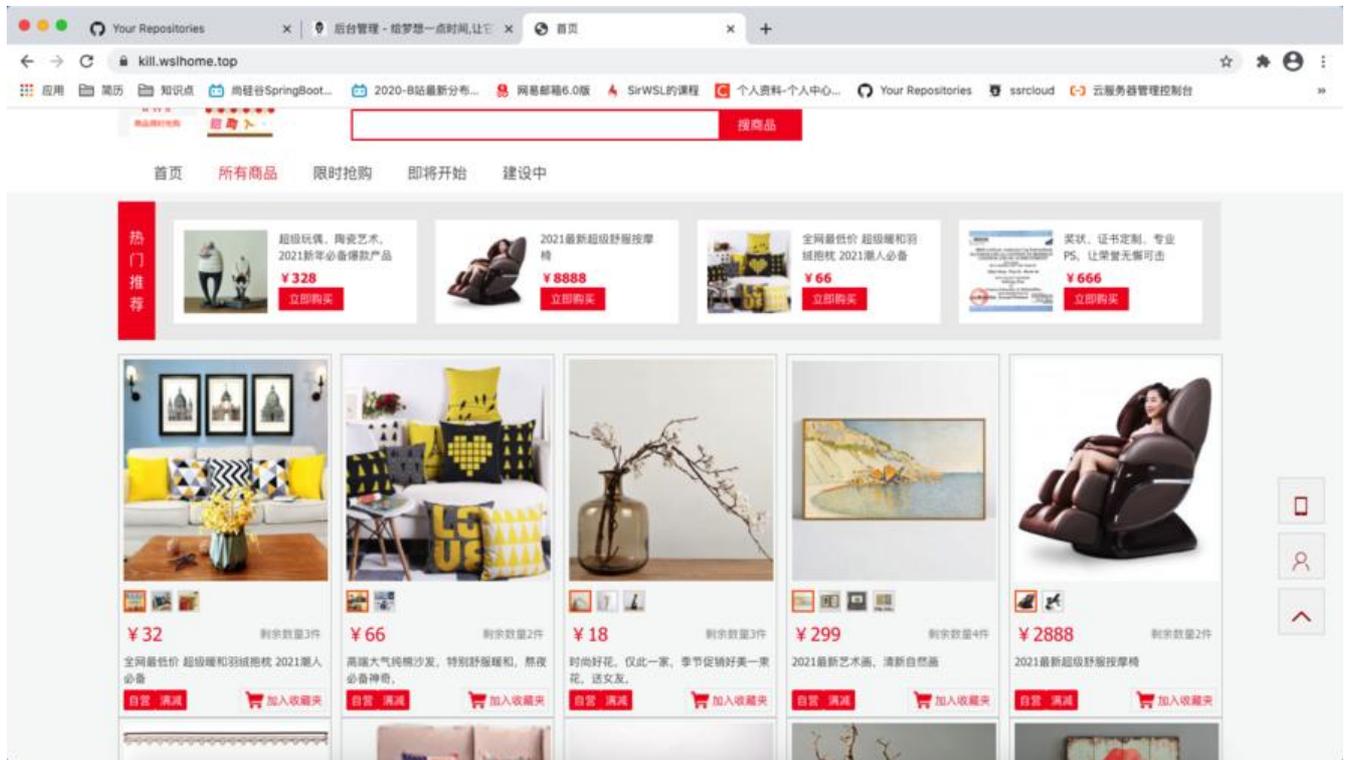
原文链接: <https://ld246.com/article/1610522437550>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 故事

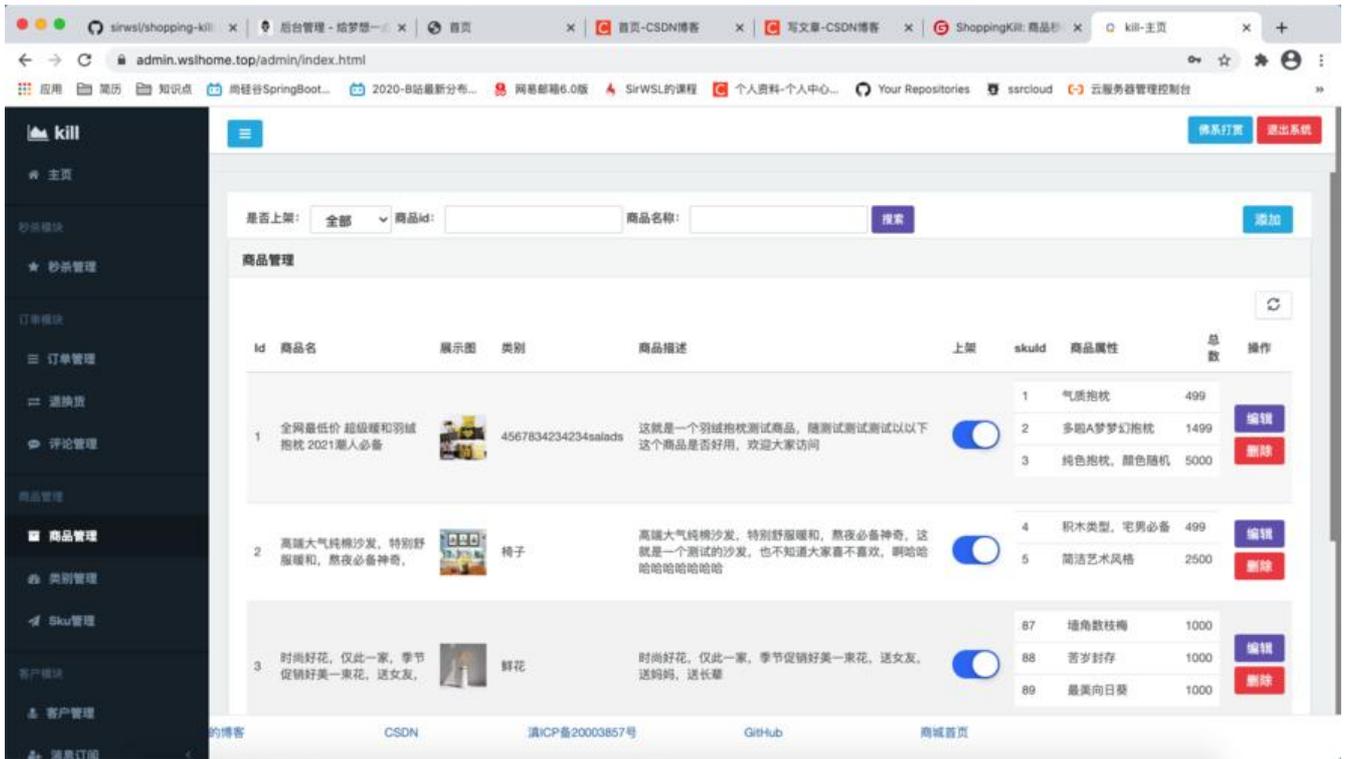
好久都没有写博客了，最近博客没写，基础知识没有进步，很多东西也没有深入去学，而自己的代码操能力与思维到时进步不少，为什么呢，因为自己一直在写毕业设计项目，本来自己只是想写一个秒杀项目，结果由于秒杀只能算是一个模块，到时候毕业答辩好像不是那么容易，然后一鼓作气，好小子把后台管理，前端商城全给写了，硬是因为一个模块写了一整套的商城系统（也不能说是一整套，还支付模块没有写，由于个人无法回调接口）。事大概就这么一个事。



现在这系统算是V1.0基本结束了，应付毕设应该没什么问题了，接下来就是写写论文，然后去学习心东西，自己应该是不会再去动它了。毕竟在这套系统上花了很多时间，也学了一些东西，现在呢就是好学习那些基础知识和看看那些源码。

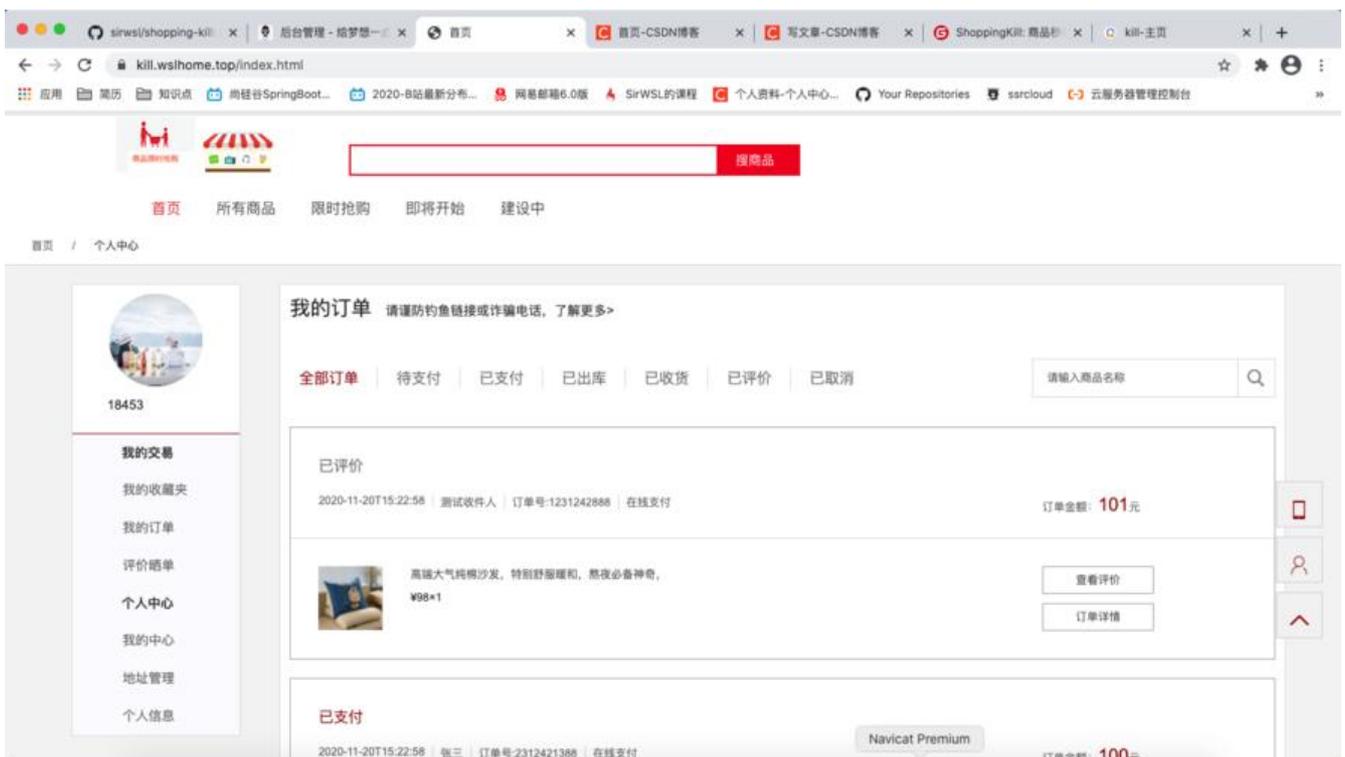
## 说说应届毕业生

作为一个应届毕业生，我真的无力吐槽，也深深感觉到，学习东西学校是交不了你多少的，要是有指人真的会很好。作为一个应届毕业生，虽然只是二本，但是我觉得很多一本或者985/211学校本科生乃至一些普本研究生，觉得真的，中国的高等教育，emmmmm，大家自行体会，我不过多评价。



不评价别人就来说说自己吧，自己之前学的成绩其实也是就一般，但是自我感觉动手能力不错，从国家级、到院系级奖状拿了30来个，然后专利、立项加起来也7、8个吧，然后成绩中上，也就是这样的我nnd，到了大二结束也就是2019.6的时候我只是直到servlet、JSP，然后会用java写一些很小的算法然后哎，我就不过多说了，现在一回忆，觉得自己大学学了个屎poop啥也没有学会。

然后到现在的springboot、锁机制、缓存、数据库、中间件、多线程全TM是2019.6以后学的。（里不是鼓励大家不学习，学校的知识也很终于）虽然说学校没交什么实际性的，但是为什么很多企业人很少有非科班的，有些企业甚至排斥培训班出来的，这就是大学课程的必要性，他会将好多基础知识原理交给你，而这些东西在实操的时候会让你更加炉火纯青。



其实在高校环境是参差不齐的，就拿毕设来说，我的系统已经完工，有的还在开题，所以明白自己想什么，然后朝着这个方向努力，很多企业招人时候指出要双一流或者985/211，这其实没有错，但是然学校一般，并不代表人家就会否定你，但是不可否认中间确实存在很大的差别，因为我之前面过BA还有很多的厂、随后虽然凉了。有的死在二三面有的死在五六面，但这很正常，毕竟差距确实存在人家加面1、2面怀疑一下能力很正常，但是也就此其实与很多人比起来自己其实打败了很多人，但也实存在差距。

这就是为什么自己写这系统加入了很多东西：

- springboot+Mybatis-plus搭建
- redis、zookeeper、mq集群
- redis缓存、jetCache缓存
- RabbitMq消息通知、死信、延迟队列使用
- zookeeper分布式锁
- 自定义注解
- 统一封装返回
- 切面使用
- 设计模式使用
- 事物、回滚使用
- docker、nginx使用
- 图片服务器OSS使用
- stram、lambda使用
- 多线程、线程池使用
- 定时任务使用
- 短信验证、邮件服务使用
- JWT验证TOKEN令牌使用
- 雪花算法分布式算法使用
- 拦截器、Log4j日志使用
- 还有很多的工具类、组件使用
- 数据库连接池Druid使用
- MapStruct实体转化工具使用
- 然后部署时候CDN之类的使用
- ... ..

还有很多，不一一说了，之前自己想找一套系统，就像这次这样融入很多东西，但是一直没有找到，次自己写了，把能用的都用上了，也学会了很多，觉得小白或者是学生或者刚入职不久或者是一些码们可以看看，我觉得对我收获颇多，对你应该也是。

## 说说系统

这玩意，其实也不能顶太多的高并发，毕竟他只是代码实现，用的也只是java，让代码来处理高并发好像有些吃力。之前看到的是采用nginx+Lua脚本，在nginx层上就处理，然后丢入队列，如果有兴的可以自己修改，毕竟也很容易，加个队列直接掉service接口就好了，只是由于自己没接触Lua脚本

所以就没弄。

大家也可以把代码copy下来，然后自己搭建以下redis、zookeeper和图片服务器，在配置文件里面上对应内容就可以。代码仓库：GitHub：<https://github.com/sirwsl/shopping-kill>，觉得凑活的就个星吧，毕竟也不容易，代码里面每一层接口注释，和一些业务逻辑的注释打的都很全，也是为了方便大家阅读。也可以去码云上git一下，码云仓库地址：<https://gitee.com/sirwsl/ShoppingKill>

### 精选项目

自定义精选项目

#### ShoppingKill

商品秒杀系统（限时抢购系统），包含商城展示与后台管理，主要技术有：springboot、redis、zookeeper、rabbitmq、sms通知、email通知。该项目为21年毕业设计项目，其中融入基本目...

Java 👁️ 2 ☆ 4 🗨️ 0

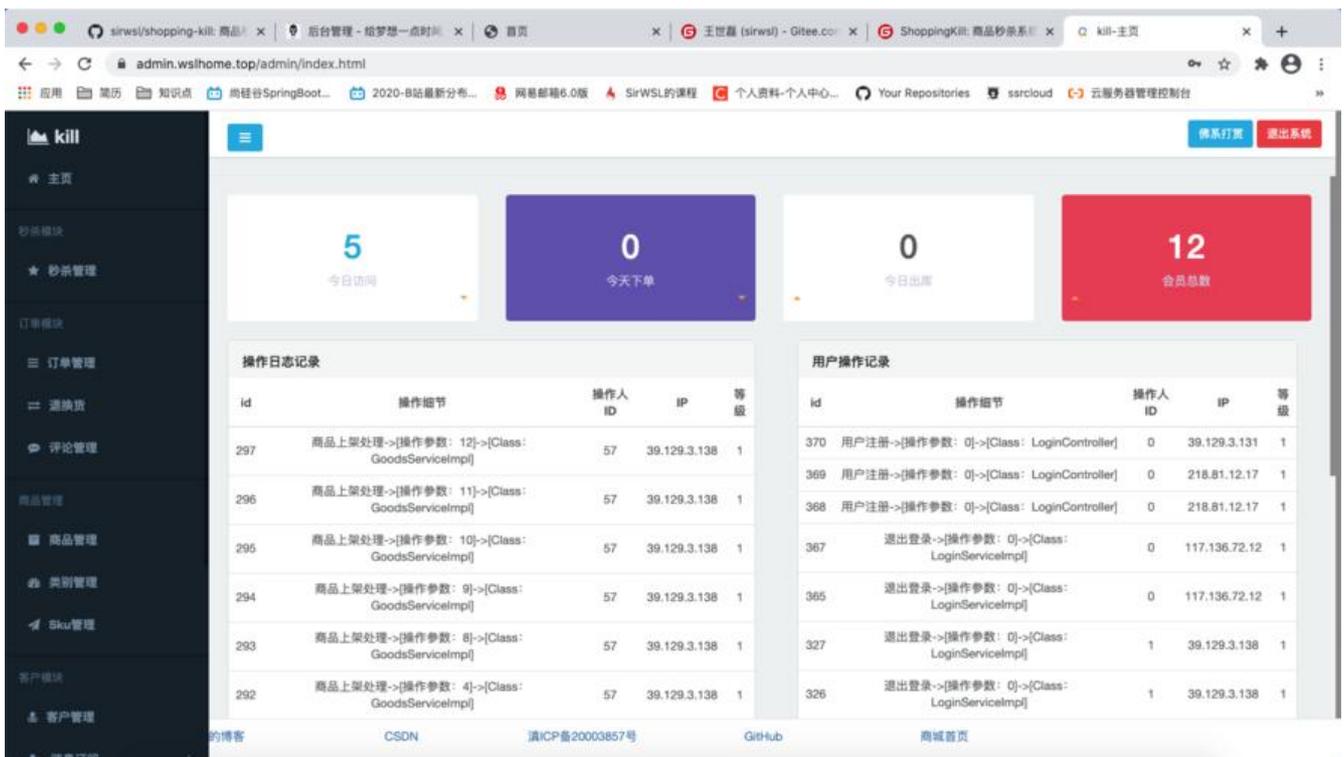
### 贡献度

2021 ▾



## 该系统适合什么样的人学习

说说该系统适合什么样的人学习吧，其实大佬都无所谓，毕竟我也只是写的毕设，和无数前辈来说，只是菜机作品，看这项目的人至少都有些java基础，因为里面用到了很多的stram和lambda表达式这些设计模式，要是没些基础，估计会很吃力。然后至少要用过SSM，知道Mybatis，里面持久层用的是Mybatis-plus，然后需要知道redis、mq之类的原理。其他的也不需要知道什么了，看代码不懂就度或者Google吧。



# 能让你收获什么

总之这玩意还是让我收获很大的，因为之前自己写demo，虽然说是前后端分离，但是并没有完全分，直到自己写这套系统时候才真正的前后端分离，有的人看到项目，可能会说，这玩意没分离，其实是自己把前端文件放在了template下面，部署时候是拿出来，打包时候也没有给他打包在一起，只因为我自己开发，图方便罢了。

然后通过这个系统收获到的：

- 明白了啥是前后端分离，如何做跨域，JWT实现令牌登录
- 自己写了结果返回统一封装
- 对异常类进行了统一封装处理
- 对部分日志采用Aspect进行存储
- 直到了redisTemplate还有一些相关的使用和配置，（之前自己只是使用jedis进行操作，收获还是大的）
- 使用了rabbitMq进行消息分发、延迟队列、死信队列等使用
- 使用了短信登录、邮件发送的一些组件
- 还有JetCache的一二级缓存使用
- 切面、自定义注解使用
- 分布式锁使用
- 明白了OSS存储、CDN转发、DockerFile的一些使用
- ... ..

还有很多就不细细说了，大家有兴趣可以自己去研究一下，再次放上项目地址

GitHub: <https://github.com/sirwsl/shopping-kill>

码云仓库地址: <https://gitee.com/sirwsl/ShoppingKill>

项目演示地址:

商城展示: <https://kill.wslhome.top>

后台管理: <https://admin.wslhome.top>

大家可以自己申请账号,查看展示效果