

个人简历

- 程绍壮
- chengsz7890@gmail.com
- 广州

工作经历

- 广州房企网络科技有限公司
 - 私企，互联网/移动互联网/电子商务行业
 - 2015.12 —— 至今
 - Java开发工程师
 - 工作职责
 - 1、根据业务需求完成功能模块设计、开发、测试、交付。
 - 2、按照功能组件的详细设计，编写规范的代码，注释。
 - 3、学习新技术框架和组件，进行评估和测试后投入使用。
 - 4、修复程序BUG、解决疑难问题。
 - 5、编写技术设计文档。
 - 6、完成上级交办的其他事项。

教育经历

- 西南大学网络继续教育学院（2017.09-2020.01）大专 | 计算机应用技术

技能标签

- Java、Spring、SpringBoot、Maven、MyBatis、Tomcat、MySQL、jQuery

兴趣爱好

- 个人对技术的理解，地基打得好、大楼盖得高。即尽可能深入浅出基础知识，然后学习新技术、框架或者语言。
- 阅读Java核心技术、Java并发编程实战、深入理解Java虚拟机等书籍、源码。对自动拆装箱、泛型、并发、反射、AOP、注解、Java8的lambda表达式有较为深入的理解和研究以及使用。
- 熟练使用Git/GitHub，对时下流行的容器化、微服务技术如: Docker、SpringBoot、SpringCloud进行研究，使用。
- 善于使用各种编辑器用于帮助提高开发效率，如: Sublime Text3、Visual Studio Code、Vim等等。
- 熟悉使用Linux(Ubuntu、CentOS)各种命令，喜欢在Windows10使用Xshell连接使用来进行使用，搭建Samba来实现Windows和Linux文件共享实现Windows系统、文件备份的存储。
- 熟练使用Windows10，喜欢编写脚本代替一些手工的操作，如开机启动时自动映射远程磁盘、打开必要的应用等等。

项目经验

- 外部商品对接项目(2016.06——至今)

- 项目负责人、Java开发工程师
- 广州房企网络科技有限公司

- 项目描述

- 外部商品对接项目是大尚国际商城的子系统，主要作用是接入第三方平台的商品进行销售，主要有商品更新对接任务和商品录入两大功能，并实现框架级定时多任务、多线程的商品对接更新和人工上传方式商品录入功能的结合。系统有controller、service、mapper、task等4层架构。基于Java1.8.x、nginx1.12.x、SpringBoot2.1.x、MyBatis、Thymeleaf、jQuery等技术。数据库使用Redis3.2.x、MySQL8.0.x。容器化基于docker、docker-compose，镜像仓库harbor。平台支撑方面: 开发环境基于Windows10、使用IntelliJ IDEA(IDEA)2.x以及Subversion进行开发和版本控制。生产环境基于公司内部服务器以及Windows Server、Linux操作系统。

- 项目职责

- 2020年

- 1、负责将项目以Docker容器的方式进行部署。首先准备好docker、docker-compose、docker仓库harbor环境，并对harbor进行相应HTTPS配置，学习编写Dockerfile、docker-compose文件的用于构建项目的镜像、容器以及其依赖组件（MySQL、Redis等），并相应的修改数据库URL的IP和端口为docker-compose的服务名称。然后学习使用Maven插件jib用于免于编写Dockerfile多层快速构建应用镜像，并将应用镜像推送至远程主机的docker仓库harbor中。最后登录harbor仓库拉取最新应用镜像，使用docker-compose命令行构建并启动应用容器以及其依赖。
- 2、负责使用Maven和Tomcat插件结合实现命令行一键打包和部署项目。首先在外部的Tomcat、Maven配置文件中添加远程Tomcat地址、用户名、密码以及Tomcat与Maven插件依赖。然后把MySQL连接器交给外部Tomcat的管理。封装Spring的ScheduledExecutorFactoryBean用于替换ScheduledThreadPoolExecutor对象，并重写destroy方法。用以解决Tomcat重新部署项目时MySQL连接池、ScheduledThreadPoolExecutor线程池未关闭导致内存泄漏问题。

- 2019年

- 1、负责引入Thymeleaf模板引擎, view层使用HTML替代JSP。在SpringBoot配置文件中添加thymeleaf模板文件的前缀（路径）和后缀名，根据SpringBoot文档把静态内容(js/css)存储至static目录下，并在请求过滤器适配器中将静态内容处理对象加入白名单。将JSP内容拷贝至HTML文件，根据Thymeleaf文档对文件头、文本表达式、路径格式以及包含其他文件格式等等进行相应的修改。
- 2、负责学习和使用Mybatis的分页插件pagehelper用于扩展Mybatis分页功能。首先为service层对象中的需要的方法添加分页注解、然后使用Aspectj和Component注释构建面向切面的Spring bean，并在bean方法上添加执行有分页注解方法的表达式参数的before注解。最后使用pagehelper提供的接口、分页函数，并根据输入分页参数进行相应的处理。
- 3、负责学习和使用Mybatis来代替SpringJDBC作为项目的主要数据持久层框架来解决SpringJDBC下需要在代码中使用字符串的方式拼接SQL，容易出现SQL语法错误，且IDEA也无法检测等问题。采用XML的配置方式，首先将原service接口方法拷贝至Mapper接口，创建映射命名空间对应的Mapper接口XML文件。添加ID与Mapper接口方法同名的增删改查元素，将与元素ID相同service接口实现对象方法的SQL语句拷贝到元素中。创建ResultMap元素以及对应的对象，在mybatis-config.xml中配置相应的设定、resultMap类型别名、映射Mapper接口XML文件。然后在创建SqlSessionFactoryBean对象时设置mybatis-config.xml文件的位置、数据源并创建SqlSessionFactory。最后使用SqlSessionFactory作为构造参数创建

SqlSessionTemplate对象，Mapper类作为方法参数来构建Mapper对象。

■ 2018年

- 1、负责引入spring-aspects依赖用于注解的方式实现数据库操作的事务管理。添加spring-aspects的依赖的AnnotationTransactionAspect和Transactional注解实现更灵活的面向切面数据源事务管理，用于替代之前XML的实现方式。在创建数据源事务管理对象DataSourceTransactionManager后，将其注入至AnnotationTransactionAspect面向切面事务对象中，然后在需要进行事务管理的子类对象的方法上添加Transactional注解，并输入相应的数据源事务管理对象bean名称、布尔类型只读等参数。
- 2、负责引入新框架升级项目架构。添加SpringBoot用于升级和替换Spring4.0.x来实现项目的简化、自动化配置、使用内置Tomcat即可启动，并以Maven来实现项目的构建、组件依赖管理。结合SpringBoot与Maven实现默认、自定义项目环境配置、打包。通过pom、yml文件等一系列配置后，在进行项目打包时，Maven根据给定的值或默认值修改SpringBoot时默认配置文件的spring.profiles.active属性值。在项目启动时SpringBoot除了加载默认配置文件，还会根据spring.profiles.active属性加载对应的配置文件。
- 3、负责升级Java运行环境和集成开发环境。将原有的JDK1.6.x升级为1.8.x，相应的Tomcat6.x升至Tomcat9.x。由于当前的MyEclipse10.x亦不支持JDK1.8.x，经过学习和使用高版本的MyEclipse和IntelliJ IDEA开发工具，决定将该项目迁移至IntelliJ IDEA。

■ 2017年

- 1、负责封装构建动态方法同步工具类。使用CGLib Nodep2.1_3(AOP框架)的MethodInterceptor接口以及Enhancer对象和同步注解组合实现多线程商品更新对接任务中更灵活的方法同步、以及方法同步必要的控制台日志输出。期间解决了Enhancer构建被代理对象时抛出NoSuchMethodException，原因是其在使用反射的方式构建对象前会根据参数类型数组查找到相应的构造器，如果输入参数类型为基础数据类型此时已经转换为其(引用数据类型)封装对象，就可能会因为参数类型差异产生无法找到相应的构造器。解决办法是让构造器参数类型都使用引用数据类型（即其封装对象），无论使用new关键字或者反射构建来对象无需担心构造器参数类型问题。
- 2、负责将商品更新对接任务升级为定时多线程的处理方式来提高数据更新效率问题。通过学习和使用ScheduledThreadPoolExecutor(替换Timer)、ThreadPoolExecutor、Runnable、Condition、Atomic对象、队列存储数据和更新对接任务集合，基于接口标准化、生产消费模式的方式实现多线程商品更新对接实现。在更新对接任务执行过程中对抛出的异常中进行标识和统计以及持久化状态位、强制关闭线程池进行同步操作来中断任务，并以可控的方式重新执行更新对接任务的异常处理机制，来处理执行更新对接任务过程中出现的对接服务器或者数据的异常问题。
- 3、负责与同事一同解决定时商品对接更新和商品录入同时操作可能产生的问题。由于商品对接更新是程序操作，而商品录入则是人工操作，由于他们无法进行协调先后顺序，加上商品有多仓库库存的因素，商品对接更新采用替换(删除后插入数据)的方式更新库存，若商品录入同时操作可能会导致库存数据关联或丢失等问题。经过商讨，通过添加中间表并设置逻辑删除标志位、错峰更新等方式进行解决。

■ 2016年

- 参与外部商品对接管理的设计、开发、测试。主要负责商品更新对接任务，根据对接文档设计，基于HttpClient编写程序请求和存储新、旧商品接口的数据。并将旧商品热数据与生产环境的商品关联后，将其接入生产环境进行更新，并根据业务需要，使用Timer以及封装TimerTask对象实现商品对接更新任务定期执行一次。

● 大尚国际商城（2015.12——至今）

- Java开发工程师
- 广州房企网络科技有限公司
- 项目描述

- 大尚国际商城是一个批发零售一体化的平台。商城系统主体结构由前台、后台、商家后台以及其它服务系统

组成。系统有controller、service、dao等3层架构，基于Jdk1.6.x、nginx、Tomcat6.x、Spring4.x（数据持久层使用SpringJDBC）、JSP、jQuery、Vue等技术。缓存/数据库使用Memcached1.2.6、MySQL5.x。平台支撑方面: 开发环境基于Windows、使用MyEclipse10.x以及Subversion进行开发和版本控制。生产环境基于云服务器以及Windows Server操作系统。

◦ 项目职责

■ 2019至2020年

- 1、协助同事解决跨域登录无法请求数据问题。通过使用Memcached替代Session存储用户登录信息方案，帮助开发同事解决特殊情况下，需要从商城前台往后台跨域登录成功却无法再请求活动数据的问题。原因是跨域的每个请求会生成一个Session导致登录信息丢失。在开发和测试环境没发现问题，是因为和生产环境不一致导致的。生产、测试环境的Tomcat和项目比例是1:N，而生产环境则是1:1（为了杜绝再出现类似问题，开发、测试环境尽量与生产环境一致）。经过验证，在后台系统登录功能、请求拦截器等两个请求点的SessionID它们都是唯一的。
- 2、协助同事解决SQL查询效率问题。通过使用MySQL的EXPLAIN关键字对开发同事的复杂SQL进行解析，解决了困扰他一个上午的问题。原因是SQL中条件字段没有使用或添加索引进而全表扫描，加上循环查询导致效率越来越低。
- 3、协助同事将第三方电子合同SDK从1.8.x降至Java1.6.x。难点在于使用jd-gui反编译为Java源码后，某些方法会有代码的语法错误，需要根据当前代码上下文进行相应的修改，且需要Java1.8.x的集成开发环境打开对应的class文件，检查两者间的方法功能是否有实质性的改变。
- 4、负责重写文章首页、文章列表的分页组件，提高页面美观、以及使用体验。

■ 2018年

- 1、负责商城前、后台以及PC、移动端文章板块的改版、以及优化，文章首页的最新文章版块实现随机显示近3天。文章详情添加推荐商品，随机推荐当前分类下文章、关键词链接。通过碎片化缓存和对象存储加上CDN技术实现文章首页、详情的毫秒级响应。
- 2、参与商城后台活动管理的设计、开发、测试。主要负责砍价活动主题、砍价活动用户、单品优惠券、商品闪购标签功能模块，对原有的商品、品牌闪购等活动模块追加关联商品功能。

■ 2017年

- 1、负责解决Apache POI导出大数据量的效率问题。通过查询官方文档，多方测试，将Apache POI库正在使用的API XSSFWorkbook 替换为SXSSFWorkbook，其通过流的方式来提高内存和磁盘的IO效率，即默认内存每加载101条数据，就写入100至磁盘上。以此解决了使用Excel导出大数据量的效率问题。
- 2、参与商城前台、后台系统的文章业务的设计、开发，测试。主要负责文章关键字、文章专题图片、文章友情链接、文章分类、文章评论等模块。
- 3、负责商城后台系统的商品详情管理下好货推荐模块的开发、测试。

■ 2015至2016年

- 1、参与商城后台商品价格、库存管理的开发、测试。主要负责价格、库存数据的Excel导入、导出等功能模块。
- 2、负责商城前台系统手机端的首页商品懒加载显示、分类菜单等功能开发，测试。
- 3、参与商城后台系统的浏览量管理的开发、测试。主要负责页面浏览量、推广浏览量、商城浏览量等功能模块。