

## 1. 个人介绍

面试官您好 我叫雷碧玉来自于广西桂林毕业于xxx学校，目前有2年工作经验了，之前有走过2个项目，分别是瑞吉外卖和在线销售系统。最近一个项目是瑞吉外卖，该项目主要从事的业务是餐饮企业。项目主要采用xxx技术架构，项目分 前台web用户端 web后台 app客户端 app后台管理等 我在这个项目中承担了xxx xxx xxx 等模块的开发，主要用到的技术有xxx xxx xxx 我的自我介绍完毕了。

## 2. 在线销售系统介绍

我最近的一个项目是在线销售系统，该项目提供的服务主要是辅助企业的销售人员对不同类型的客户进行筛选和有效沟通的。

我在这个项目中负责的项目有

### 1. 日志记录功能

用到的技术主要是：使用Spring的AOP+自定义注解

流程：1自定义注解（@Log注解）--2编写日志处理类--3异步方式存储日志--4业务方法调用日志。

自定义注解：模块、功能、操作人类别、是否保存请求的参数

优势：代码独立于其他业务逻辑代码，不仅实现了解耦，而且避免了冗余代码。

### 2. 数据权限与角色进行绑定功能

使用的技术主要为：Spring的AOP+自定义注解

流程：1利用sql实现数据权限--2自定义数据权限注解@DataScope--3利用aop实现该注解--3.1确定切入点--3.2利用通知类型拼接sql

该怎么做好：

明确controller接口的访问权限，不同的人分配不同的角色。如果遇到一个功能是多个角色都可以访问，入口都是同一个controller，要求返回的数据不一样时，由业务层的逻辑代码来控制。

## 3. 批量线索导入

技术：使用EasyExcel

easyExcel操作Exel文档。

读excel方式：

- 创建实体类（属性与excel列映射 @ExcelProperty)

- 创建监听器,invoke方法, 每读取一条记录时, 就执行一次
- 调用EasyExcel.read(文件, 实体类, 监听器).sheet().doRead()
- 优化: 批量读取: 不一一条一条操作, 累加起来达到一定数量后再操作; 【注意】: 在最快的方法中 再次调用一下。批量值为5000为佳。
- 介绍mybatis的批量操作, sqlSession.openSession(ExecutorType.BATCH)
- 多线程 批量处理

## 4.商机

## 5.合同上传

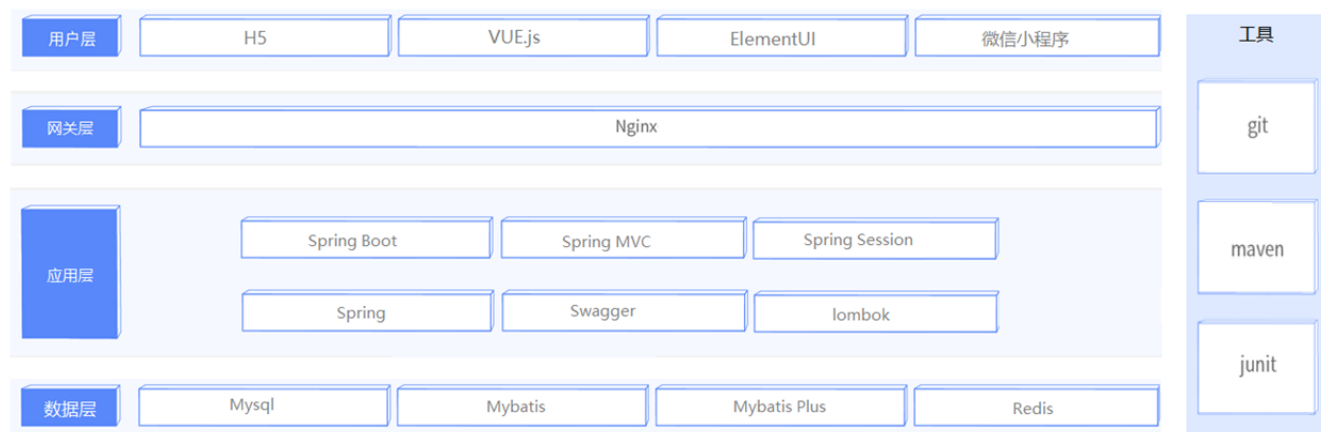
技术: 使用Minio实现

自己部署minio服务器(保存敏感, 企业中重要的文件)

使用它的minio客户端上传文件(bucketName, 文件名, 文件流)

## 2.瑞吉外卖

该项目主要从事的业务是餐饮企业



### 1). 用户层

本项目中在构建系统管理后台的前端页面, 我们会用到H5、Vue.js、ElementUI等技术。而在构建移动端应用时, 我们会使用到微信小程序。

### 2). 网关层

Nginx是一个服务器, 主要用来作为Http服务器, 部署静态资源, 访问性能高。在Nginx中还有两个比较重要的作用: 反向代理和负载均衡, 在进行项目部署时, 要实现Tomcat的负载均衡, 就可以通过Nginx来实现。

### 3). 应用层

SpringBoot: 快速构建Spring项目, 采用 "约定优于配置" 的思想, 简化Spring项目的配置开发。

Spring: 统一管理项目中的各种资源(bean), 在web开发的各层中都会用到。

SpringMVC: SpringMVC是spring框架的一个模块, springmvc和spring无需通过中间整合层进行整合, 可以无缝集成。

SpringSession: 主要解决在集群环境下的Session共享问题。

lombok: 能以简单的注解形式来简化java代码, 提高开发人员的开发效率。例如开发中经常需要写的javabean, 都需要花时间去添加相应的getter/setter, 也许还要去写构造器、equals等方法。

Swagger: 可以自动的帮助开发人员生成接口文档, 并对接口进行测试。

#### 4). 数据层

MySQL: 关系型数据库, 本项目的核心业务数据都会采用MySQL进行存储。

MybatisPlus: 本项目持久层将会使用MybatisPlus来简化开发, 基本的单表增删改查直接调用框架提供的方法即可。

Redis: 基于key-value格式存储的内存数据库, 访问速度快, 经常使用它做缓存(降低数据库访问压力, 提供访问效率), 在后面的性能优化中会使用。

#### 5). 工具

git: 版本控制工具, 在团队协作中, 使用该工具对项目中的代码进行管理。

maven: 项目构建工具。

junit: 单元测试工具, 开发人员功能实现完毕后, 需要通过junit对功能进行单元测试。

### 1.维护餐厅中的套餐信息, 包含查、增、改、删、启售、停售等功能

### 2.维护用户在移动端下的订单信息, 包含查、取消、派送、完成

### 3.订单报表下载等功能

## 3.熟悉技术

- ① 熟悉SpringBoot, SSM, Mybatis, SpringMVC等开源框架;
- ② 熟练使用Idea、Git、Navicat、Maven等管理工具;
- ③ 熟悉使用Redis缓存数据库;
- ④ 熟练操作Linux命令, 了解Docker容器;
- ⑤ 熟悉JWT安全验证等。

## 1.使用Spring框架的好处

### 使用Spring框架的好处是什么？

#### 问题回答

1. 轻量:Spring 是轻量级, 无侵入。
2. 控制反转 (IOC) :Spring通过控制反转实现了松散耦合, 对象们给出它们的依赖, 而不是创建或查找依赖的对象们。
3. 面向切面的编程(AOP):Spring支持面向切面的编程, 并且把应用业务逻辑和系统服务分开。
4. 容器: Spring 包含并管理应用中对象的生命周期和配置。
5. MVC框架: Spring的WEB框架是个精心设计的框架, 是Web框架的一个很好的替代品。
6. 事务管理: Spring 提供一个持续的事务管理接口, 可以通过配置文件或者注解轻松实现。
7. 异常处理: Spring提供了全局异常处理功能, 只需要声明一个全局异常处理器就可以捕获所有异常信息。

## 2.Spring用到的设计模式

### spring用到的设计模式？

#### 问题回答

1. 工厂设计模式: Spring使用工厂模式通过 BeanFactory、ApplicationContext 创建 bean 对象。
2. 代理设计模式: Spring AOP 功能的实现。
3. 单例设计模式: Spring 中的 Bean 默认都是单例的。
4. 模板方法模式: Spring 中 jdbcTemplate、hibernateTemplate 等以 Template 结尾的对数据库操作的类, 它们就使用到了模板模式。
5. 包装器设计模式: 我们的项目需要连接多个数据库, 而且不同的客户在每次访问中根据需要会去访问不同的数据库。这种模式让我们可以根据客户的需求能够动态切换不同的数据源。
6. 观察者模式: Spring 事件驱动模型就是观察者模式很经典的一个应用。
7. 适配器模式: Spring AOP 的增强或通知(Advice)使用到了适配器模式、spring MVC 中也是用到了适配器模式适配Controller。

## 3.Spring中AOP底层的实现基于动态代理

### 3.1JDK的动态代理-bean没有实现接口

### 3.2cglib-bean实现接口

Spring中AOP底层的实现是基于**动态代理**进行实现的。

常见的动态代理技术有两种：JDK的动态代理和CGLIB。

两者的区别如下所示：

- 1、JDK动态代理只能对实现了接口的类生成代理，而不能针对类
- 2、Cglib是针对类实现代理，主要是对指定的类生成一个子类，覆盖其中的方法进行增强，但是因为采用的是**继承**，所以该类或方法最好不要声明为final，对于final类或方法，是无法继承的。

Spring如何选择是用JDK还是cglib？

- 1、当bean实现接口时，会用JDK代理模式
- 2、当bean没有实现接口，会用cglib实现
- 3、可以强制使用cglib

- 在springboot项目可以配置以下注解，强制使用cglib `@EnableAspectJAutoProxy(proxyTargetClass = true)`

## 4.Spring注解：@Aspect, @Before, @After, @Around, @Pointcut - 用于切面编程 (AOP)

## 5.说一下spring的AOP

## ssm-说一下spring的AOP?

### 问题回答

1. AOP，也就是**面向切面编程**；其次还有**面向过程**和**面向对象编程**。
2. Spring AOP实现主要是**通过代理类**的方式实现，有**jdk动态代理**和**CGLIB代理**两种方式。
3. Java动态代理**只能够对接口**进行代理，不能对普通的类进行代理（因为所有生成的代理类的父类为Proxy，Java类继承机制不允许多重继承）；CGLIB能够**代理普通类**；
4. Spring内部，发现如果目标对象没有实现接口，则默认会采用CGLIB代理；如果目标对象实现了接口，默认会采用Java动态代理；
5. AOP用到的动态代理是对**方法增强**的方式之一。
6. AOP的典型应用就是**事务管理**。