1. 中控的经历

    - 入职交通部门，做交通部门核心的软件产品信号控制系统的开发，这个系统主要是给各个城市的交警部门来做交通管控的，滨江、萧山、余杭、绍兴柯桥都用的这一套系统，主要功能包括

        - 是对路口的信号机、红绿灯、摄像设备、雷达设备等进行实时的控制、信号采集、监控和异常报警等

    - 我的主要贡献

        - 一是这个项目比较老，一开始是一个大型的单体项目，我参与到了系统的垂直业务划分，逐步将整个大单体应用逐步微服务化
        - 二是对算法推送的红绿灯控制方案实时发布到实际的路口信号机上，实现不同时段的路口红路读秒方案的动态调整，实现高峰期交通拥堵的合理优化
        - 三是做一些重大节日或重要领导到来时对路口的红路灯设备进行一些特殊的管控功能，比如给领导开出一条交通信号灯上的绿色通道
        - 二是对滨江余杭萧山等地部署做本地的定制化服务
        - 三是和阿里做了郑州城市大脑项目的交通板块，在信号控制系统定制化的基础上，做了一些衍生系统的研发，比如针对阿里的交通算法提供一个仿真演示的平台，实现对区域路口的交通拥堵的预测治理演示；实现全市电动车历史轨迹数据监控的平台，从阿里的大数据平台中获取到电动车的轨迹数据进行分析，然后调整信号及的红绿灯方案，从红绿灯的读秒上对电动车行车安全行车进行优化，避免事故的发生
        - 这个项目我是作为项目经理，同时也进行相应的开发部署工作，主要是对公司自研的红绿灯控制系统进行一些定制化的改造，配合郑州交警做了一些定制化的功能，比如监控全城的不同时段的机动车流向及路口拥堵状态，接入算法修改路口的红绿灯读秒方案改进高峰期的读秒方案，监控电动车轨迹统计等功能，对电动车行车安全进行分析与干预
        - 根据功能制定详细的测试用例表，进行详细的测试

2. 丁香园的经历

- 入职的是大数据部门，主要的工作是参与大数据核心产品odep的web开发与迭代，同时负责周边的一些工具性web应用的开发

- odep底层是基于spark进行了封装，形成了自己的一套sql语法，称之为mlsql，实现通过在页面上编写sql脚本来实现数据的查询分析，同时也支持定时任务来执行sql实现后台数据的分析，支持针对sql、python、以及一些shell脚本的定时任务的执行

- 我的主要工作

- 一是对数据访问权限的改造，开发了基于spring security + oauth2.0协议的用户权限管理系统RMS，并与公司的OA系统进行打通，实现在OA系统中走OA审批申请申请需要库表的访问权限

- 参与数据中台的功能设计和实现，首先是针对页面上提交sql到底层的mlsql服务器的流程设计（首先根据提交的sql从mlsql服务器的sql分析服务器拿到sql中涉及到的库表及字段信息，然后从用户管理平台获取到用户的库、表、敏感字段的权限信息，对比用户是否有查询权限，无权限则拒绝执行sql）

- 二是参与大数据核心平台数据中台的功能开发，实现数据自助分析、数据表单查询、任务管理等功能的设计与实现

- 参数脚本管理、自助分析、表单查询、数据血缘

- 三是在参考kibana实现了一个表单查询的功能，原来用户sql脚本的基础上提供一些通过填写表单来给sql提供查询条件查询分析数据的功能
- 四是对调度任务管理的完善，调度系统是公司基于yarn开发的merlion调用系统，针对merlion提供的调度api接口，我实现了merlion-sdk的包，将调度管理配置由odep来管理，将用户编写的sql脚本及调度配置通过merlion-sdk提交给merlion，实现调度功能的闭环
- 五是根据业务部门反馈查看数据报表的需求编写sql，开发调度程序
- 另外对web项目中用到的一些基础库进行了封装，比如实现了一个声明式的接口调用工具包、利用spring提供的扩展点扩展功能，比如入参出参的绑定，实现接口调用自动日志审计工具，redis操作的工具类的封装
- 六配置中心引入，调研配置中心apollo和nacos，由于重点在于配置的安全性，不需要使用服务发现与注册这些功能，最终考虑选择apollo作为配置中心
- 使用spring 提供的拓展对接口的入参、出参进行统一的封装，对一些日志审计工具进行封装 选用apollo的理由：
    1. 配置的安全性：按namespace的粒度来管理配置，同时区分配置的查看、修改、发布权限
    2. 不同环境的配置：区分开发环境、测试环境、uat环境、生产环境
    3. 支持负载均衡
    4. 支持灰度发布
- 开发monitor监控告警系统对一些告警数据进行聚合分发
- rms统一用户管理平台的开发，基于spring security+oauth2.0协议实现了对用户的管理、登陆、授权等服务操作，实现了大数据所有web应用用户权限的统一管理，对接OA，将rms作为企业微信的一个接入应用，在大数据应用内部，又将rms作为授权认证中心
- 负责一些项目的code review工作

rms平台 rms平台登陆认证使用的是spring +oauth2.0协议实现登陆授权操作 rms的主要亮点： 1. 用户登陆后提供一个应用大厅界面，通过大厅界面可以访问接入rms-auth的所有应用 2. 引入企业微信扫码认证，与OA打通绑定 3. 使用spring security + oauth2.0协议的授权码模式实现用户的登录认证及授权访问 4. 提供api针对各个系统中无权限访问数据时提供便捷权限申请的途径

为什么没有使用jwt 作为token -- 只需要在web端进行登陆访问，没有其他客户端的需求 jwt的数据量较大，我们这边有些用户的权限资源信息会比较多，传输起来可能会有一定的负载

用户登陆流程 用户在浏览器端输入https://rms.k8s-test.uc.host.dxy 访问rms rms前端页面校验当前域名下是否有token，当没有token时，跳转到扫码登陆页面，扫码登陆的码是由企业微信提供 用户通过企业微信扫码并授权登陆 企业微信在验证通过后回调rms的/scan接口，并提供一个ticket票据信息，票据信息中包含了用户的企业微信名、企业邮箱等信息 rms通过企业微信提供的sdk包解析票据中包含的用户信息，拿到用户名和邮箱 rms对比用户名是否在rms用户表中合法，合法则走ouath2.0的登陆认证流程 rms-web将自身作为rms-auth（认证授权中心）的一个子平台，将自身的clienId、ClientSecret发送给rms-auth，rms-auth认证客户端合法后发送一个code码给rms-web rms-web拿到code码后再次向rms-auth 发起请求拿到token 至此登陆认证流程结束，rms-web将token返回给前端页面并重定向到首页

首先我们在前端页面嵌入企业微信的扫码页面，用户扫码完成后，在企业微信中点击确认登陆，这个操作相当于企业微信给rms授权，授权完成之后企业微信会回调rms这边提供的接口，并提供一个token参数，rms拿

到token之后，再通过token调用企业微信提供的获取用户信息的接口，拿到用户信息后，跟本地数据库中的用户信息进行对比，如果能符合，则登陆成功

工作中的亮点，难点：

1. 新封装一套基础的jar包，解决之前同事提供的jar包中各种版本冲突导致的各种问题
2. 对配置中心的选择和搭建，考虑到项目中的实际需求以及需求，选择apollo作为配置中心，一是因为apollo的配置中心支持权限控制，二是各个项目的控制就是以项目来进行划分，比较清晰，同时也可以在各个项目配置中划分namespace，按照namespace来进行权限控制，三是在项目中使用起来比较简单
3. 业务上的创新，针对业务部门运营同事不会编写sql的 工作中多线程的例子 利用多个线程对数据库中大量的数据进行数据转换然后存储 jdk 8 jdk 11新特性 stream编程 工作中进行java调优的例子 工作中具体的sql优化例子

在有大量写操作的redis实际生产环境中，如何才能保证redis数据的一致性和redis的读写效率

如何做负载均衡 redis的key过期是如何实现的

spring security 的具体工作原理： Spring Security会在Web应用程序的过滤器链中添加一组自定义的过滤器，这些过滤器可以实现身份验证和授权功能。当用户请求资源时，Spring Security会拦截请求，并使用配置的身份验证机制来验证用户身份。如果身份验证成功，Spring Security会授权用户访问所请求的资源。

1.用户请求Web应用程序的受保护资源。

2.Spring Security拦截请求，并尝试获取用户的身份验证信息。

3.如果用户没有经过身份验证，Spring Security将向用户显示一个登录页面，并要求用户提供有效的凭据（用户名和密码）。

4.一旦用户提供了有效的凭据，Spring Security将验证这些凭据，并创建一个已认证的安全上下文（SecurityContext）对象。

5.安全上下文对象包含已认证的用户信息，包括用户名、角色和授权信息。

6.在接下来的请求中，Spring Security将使用已经认证的安全上下文对象来判断用户是否有权访问受保护的资源。

7.如果用户有权访问资源，Spring Security将允许用户访问资源，否则将返回一个错误信息。

6.Spring Security基于用户名和密码的认证模式流程？ 请求的用户名密码可以通过表单登录，基础认证，数字认证三种方式从HttpServletRequest中获得，用于认证的数据源策略有内存，数据库，ldap,自定义等。

拦截未授权的请求，重定向到登录页面的过程：

当用户访问需要授权的资源时，Spring Security会检查用户是否已经认证（即是否已登录），如果没有登录则会重定向到登录页面。

重定向到登录页面时，用户需要输入用户名和密码进行认证。

表单登录的过程：

用户在登录页面输入用户名和密码，提交表单。

Spring Security的UsernamePasswordAuthenticationFilter拦截表单提交的请求，并将用户名和密码封装成一个Authentication对象。

AuthenticationManager接收到Authentication对象后，会根据用户名和密码查询用户信息，并将用户信息封装成一个UserDetails对象。

如果查询到用户信息，则将UserDetails对象封装成一个已认证的Authentication对象并返回，如果查询不到用户信息，则抛出相应的异常。

认证成功后，用户会被重定向到之前访问的资源。如果之前访问的资源需要特定的角色或权限才能访问，则还需要进行授权的过程。

Spring Security的认证流程大致可以分为两个过程，首先是用户登录认证的过程，然后是用户访问受保护资源时的授权过程。在认证过程中，用户需要提供用户名和密码，Spring Security通过UsernamePasswordAuthenticationFilter将用户名和密码封装成Authentication对象，并交由AuthenticationManager进行认证。如果认证成功，则认证结果会存储在SecurityContextHolder中。在授权过程中，Spring Security会检查用户是否有访问受保护资源的权限，如果没有则会重定向到登录页面进行认证。

面试中的常见问题

1. 自我介绍 --介绍工作经历、项目经历（将项目经历的主次分清楚，明确自己主要做了哪些工作）
2. 工作中的亮点，难点 （一般是说的项目上的难点、亮点）地层依赖包的封装，解决重复代码在各个项目中粘贴复制的问题，业务上提出创新，解决其他不会sql的同事的sql查询的问题，调研配置中心，最终选定使用apollo作为配置中心
3. 如何排查线上的问题（常规方法，遇到特殊的日志问题时的排查）
4. 如何做线上应用的性能优化
5. 数据库表设计方面的知识
6. 最近有没有了解过一些新的技术、有什么收获
7. 常规的技术问答

我的技术栈：