面试总结（黄金版）

不要照搬来背，翻译成自己的话（像我者生，学我者死）

自我介绍1：（尊贵版）--18k以上

你好，我叫java，上家公司是软通动力，工作到现在有4年了，主流技术和框架都用过，
最近做的是一个APP 项目。
项目当时分为APP端，server端 ，还有个PC 版的CMS运营中心这三块。
（考虑到手机APP体积过大）手机版主要依赖html5+android或者html5+ios 来实现，
server端当时采用springboot +dubbo+zookeeper 框架，主要为前端提供数据，和接收更新，当时采用restfull风格编程。
运营中心主要是对这两个APP 的内容进行数据维护，行为分析。

server端是一个web接口项目，由于牵扯到大用户量的频繁访问问题，为了避免项目后期项目越拖越慢，所以从架构思想上讲，整个项目能静态化的都静态化，不能静态化的全部走缓存。还有后期运营如果成功的话，访问量很有可能进一步翻倍增加，所项目直接融入了dubbo 的分布式模式，方便后期横向扩展，直接增加生产者和消费者的个数。
整个项目做的是多消费者部署，用redis 实现session 共享。
这3块项目用的是同一组数据库，当时配置的是一个mysql的集群。
手机的检索我搭了一个独立的检索服务器用的是solr。
项目中的redis缓存用的是 redis 集群也配置了哨兵模式。
对于数量关系单一的数据，像点赞，发布评论，收藏都用的mongdb 来存储，降低关系数据库的负担。
点赞我们分为新闻点赞、评论点赞，评论中还有举报、回复等功能
视频的存储当时自己搭建的FTP加nginx存的。
行为统计我们当时使用的kafka来完成。
首页轮播我们考虑到图片加载速度慢的问题，把图片进行了纵切割，加载的时候看似是一张图片，其实是一组<a>标签组合到一块的结果

我们的项目业务模块有很多：
像手机端上包括我的频道、首页轮播、热荐、新闻搜索、发布评论、点赞、收藏、分享、新闻资讯、热播综艺、荔枝视频、热门、滚动、民生、直播电台、栏目推荐、个人中心、活动广场、我的荔枝树、荔枝商城、我的收藏、行为统计等一系列功能，我的频道有很多分类，诸如：头条、推荐、国内、国际、原创、娱乐、体育等等功能模块
PC 运营端主要涉及系统管理、个人中心、频道维护、轮播管理、热荐维护、公告管理、栏目管理、新闻管理、评论管理、视频管理、活动管理、反馈管理、统计管理（日活、月活、累计用户、android下载量、ios下载量），行为分析等

cms运营中心用的是ssm框架来实现。数据库用的mysql读写分离，数据源配置的是spring aop 实现的一个动态数据源，主要就是前置通知中根据方法名前缀进行判断，注入不同的datasource数据源，所有的修改走主库，所有的查询走从库。
项目中的热荐、首页轮播等经常展示的都用redis做了缓存，我们的缓存更新策略是实时更新，操作完数据库紧接着就更新缓存。
发布评论、点赞、收藏都用的是mongdb 来存储。
系统管理我们用的五表模型加 shiro 来完成，能精确到按钮级别。
活动管理我们采用的freemarker静态页面来实现

我在公司担任高级开发，主要负责server端以及cms运营中心的项目开发，
其中server端我负责的模块有：行为统计（kafka）、热荐（redis）、首页轮播（redis）、新闻搜索（solr）、发布评论（mongo）、点赞（mongo）、收藏（mongo）、个人中心、活动广场（redis）等
cms运营中心我负责的模块有：系统管理、热荐维护、新闻管理、评论管理、活动管理等
基本情况就这些您看你还有啥问的？

自我介绍2：（尊荣版）--15k以上

你好我叫卡夫卡，上家公司是软通动力，这几年参与过的项目呢， 有APP 的、CMS的、电子政务的、BMS、 仓储物流等等。最近我们做的项目是一个APP项目 。
项目当时分为APP端，server端 ，还有个PC 版的CMS运营中心这三块，APP端分为android 版和ios 版，
server端当时采用springboot +dubbo+zookeeper 框架，主要提供各种数据接口，或者接收请求更新数据，当时采用restfull风格编程，
运营中心主要是对这两个APP 的内容进行数据维护，行为分析。
（考虑到手机APP体积过大）手机版主要依赖html5+android或者html5+ios 来实现。
我们项目业务挺多的：
像手机端就包括我的频道、首页轮播、热荐、新闻搜索、发布评论、点赞、收藏、分享、新闻资讯、热播综艺、荔枝视频、热门、滚动、民生、直播电台、栏目推荐、个人中心、活动广场、我的荔枝树、荔枝商城、我的收藏、行为统计等一系列功能，我的频道有很多分类，诸如：头条、推荐、国内、国际、原创、娱乐、体育等等
PC 运营端：主要涉及系统管理、个人中心、频道维护、banner管理、热荐维护、公告管理、栏目管理、新闻管理、评论管理、视频管理、活动管理、反馈管理、统计管理（日活、月活、累计用户、android下载量、ios下载量）、行为分析等

cms运营中心用的是ssm来实现。数据库用的mysql 读写分离，数据源配置的是spring aop 实现的一个动态数据源，所有的修改走主库，所有的查询走从库。
项目中 热荐、首页轮播 等经常展示的都用redis 做了缓存。
发布评论、点赞、收藏 都用的是mongdb 来存储。
权限用的五表模型加 shiro 来完成能精确到按钮级别。

我在公司担任高级开发，主要负责server端的接口开发，
负责的模块有：行为统计（kafka）、热荐（redis）、首页轮播（redis）、新闻搜索（solr）、发布评论（mongo）、点赞（mongo）、收藏（mongo）、个人中心、活动广场（redis）等
基本情况就这些您看你还有啥问的？

自我介绍3：（平民版）--10k以上

经理你好，我叫XXX，之前在软通动力上班，在公司担任java开发，熟悉的东西也挺多，像行业常见的ssm、ssh框架都挺熟悉，另外对springboot、springcloud也都略知一二，数据库主要用到的就oracle、mysql这些，我之前在公司里参与过不少项目，像这种APP啦、OA啦、CRM啦都参与过，这些在简历上都有。
最近呢也是刚做完一个项目，我给介绍一下：
这个项目是给江苏省广播电视集团做的，叫荔枝新闻，是个手机APP，项目采用ssm框架开发，其中涵盖了我的频道、首页轮播、热荐、新闻搜索、发布评论、点赞、收藏、分享、新闻资讯、热播综艺、荔枝视频、热门、滚动、民生、直播电台、栏目推荐、个人中心、活动广场、我的荔枝树、荔枝商城、我的收藏、行为统计等一系列功能，我的频道有很多分类，诸如：头条、推荐、国内、国际、原创、娱乐、体育等等，咳，这个APP有安卓客户端和苹果客户端，我们项目组有两个安卓、两个ios以及三个后台，我主要是做后台开发，其中项目前期参与了需求分析、数据建模、框架选型、编写需求文档、中期负责行为统计（kafka）、热荐（redis）、首页轮播（redis）、新闻搜索（solr）、发布评论（mongo）、点赞（mongo）、收藏（mongo）、个人中心、活动广场（redis）等一系列功能的开发，另外就是封装一些工具类和基类，后期负责项目上线的相关事宜，平时也监控一下项目进度，总体上来讲，全程参与了这个项目。
我的基本情况就是这样，经理你看还有什么想要了解的。

相关解释：
1、新闻来源（你们的新闻数据是从哪来的）？--不问别说
我们网站的新闻主要来源于两个渠道，一个是媒体抓取、一个是自媒体，自媒体包括一些单位和个人

2、你们的直播数据从哪里来的？--不问别说
我们这个项目是给江苏省广播电视集团做的，公司里有固定的数据来源
如果不是给某电视集团做的，那就说：我们的数据是通过第三方渠道买来的

3、当时你们这个项目是在什么地方做的？--不问别说
当时我们是在客户现场做的，我们整个项目团队都在南京

4、你们是怎么做需求分析的？--不问别说

这个吧，每个公司应该都差不多，我们接下项目之后会开会讨论每一个功能以及该功能用什么技术，同时测试人员确认测试用例，并要求客户参与，以便确认和更改，几次会议之后，双方确认没有问题，基本就可以展开后续的工作。

5、你们数据建模是怎么做的？--不问别说

数据建模我们使用的是PD（power designer），我们创建的都是物理数据模型，规范就是遵循数据库三范式、尽量表字段不要太多、合理利用表关联关系，一般我们设计完之后会开三到五次会议，让所有开发人员确认没有问题之后，开始进入下一阶段。

6、你们公司是怎么写需求文档的？--不问别说

这个东西没什么硬性标准，就是用语言把要实现的功能描述出来，说不清的时候，最多也就是附带一张流程图（使用Visual Studio），公司里都有各种文档模板，照着往上填写就行。

以下内容可以主动介绍：

7、你们框架选型是怎么选的？

我们主要从这几个方面来进行分析，A：安全性；B：执行效率；C：灵活性；D：开发人员掌握程度；E：培训成本综合对比：springmvc基于单例设计，注解模式，基于方法传参，之前struts2已经连续爆了两个致命安全漏洞，从安全性和灵活性来看，完胜struts2，spring就不用说了，从我开发以来，几乎所有项目都会用到这个东西，根据我个人经验来看，spring就是为了整合一系列毫无关系的插件，他在项目中也就两个作用：第一，创建管理bean实例，我们的controller、service、mapper都交给了spring管理；第二，控制项目中的事务，利用spring的AOP进行事务拦截，常见的事务传播特性貌似有required、new_required、never等六七种吧，记不太清了，常用的就是required这个；另外，spring还提供了一系列的注解：@Service、@Repostory、@Autowired等，持久层这块有很多成熟的技术，从处理业务的灵活性来讲，mybatis比较适合，他能执行一系列sql语句，可以说随心所欲，我们就是这么选的框架。（如果经理听完5秒没说话，开始问他：经理，咱们公司用的是什么框架啊）

8、行为统计你们是怎么做的？

当时吧，我们做这个功能，主要是为了根据用户的行为，分析出用户群体的喜好，以便我们对产品进行更新和迭代，当时我们这个功能使用的kafka来完成，我们的项目充当producer，当用户触发每一次点击行为的时候，都会向后台发送一次请求，后台收到请求之后我们会在当前线程中开启一个新的线程用于行为的统计，在多线程中，我们会把当前请求的参数、用户ID、以及请求码封装成json字符串，同时调用KafkaProducer的send方法，把这个json字符串发布到kafka的broker中，同时，我们公司还有一个cms运营中心，那是我们kafka的Customer，会不停的到broker上取出数据并存储。

刁难问法：你们kafka有没有做集群（分布式）？--不问不说

回答：我们这个项目开发没多久，kafka相当于试用，当时并没有做分布式

9、多线程你们怎么实现的？

传统的多线程一般都是实现Runnable这个接口或者是继承Thread抽象类，我们当时采用了线程池来实现，Java提供的线程池有好几种，我知道的有CachedThreadPool（缓存线程池）、FixedThreadPool（定长线程池）、ScheduledThreadPool（定时线程池）、SingleThreadExecutor（单例线程池）这几种，我们项目中采用的是SingleThreadExecutor这个线程池，他是单例模式，只开一个线程，节省开支，我们通过调用execute这个方法来执行线程。

刁难问法：你们怎么防止线程泄露？--不问不说

回答：我们项目开发时间较短，并没有涉及到这方面

10、热荐你们是怎么做的？

这是个热门推荐系统，我们采用了redis来存储热门内容（主要就是一些热门新闻），这些热门信息在我们的redis服务中以键值对的方式来存储，以方法名+时间戳这种方式作为key，以json字符串作为value，这样把每一条热门信息单独存起来，我们的redis当时与spring做了整合，当查询时，我们会调用RedisTemplate的keys方法来列出以这个方法名为前缀的所有键，在Java中做好排序之后，再根据每一个键取出对应的值，把所有取回的值组成一个List返回给前台。考虑到某些信息可能过一段时间就不在是热门，我们给所有热门信息设置了一个月的过期时间。

刁难问法：你们redis有没有做集群？--不问不说

回答：我们当时没有做集群，就是单节点服务，我对redis了解过，主从哨兵也都知道

11、首页轮播你们怎么实现的？

（微笑）我们首先在运营中心添加轮播图，同时调用redis来存储轮播图路径，我们使用方法名+时间戳的方式作为key，json字符串作为value，APP端发起请求的时候，我们server端直接到redis上取出所有的轮播图路径返回给前台，考虑到图片过大影响加载速度，我们当时采用了图片纵切割，页面上的每一张轮播图片实际是一组<a>标签的集合。

12、你能介绍下新闻搜索吗？

这个功能我们使用的是solr检索，也是为了提高检索效率，当时我们搭建了一个单独的solr服务，并且配置了IK中文分词器，solr的核心就是索引库或者叫做core，核心文件就是solrconfig和schema这两个，项目中我们通过solrj来对

索引进行操作，当我们的一条新闻通过审核，同时我们会生成solr索引（通过addBean，还需要commit事务），并且同时生成静态页面，APP进行搜索时，server端直接从solr进行检索，当APP点击某条新闻查看详情时，直接请求静态页面。

刁难问法：你了解正排索引和倒排索引吗？

回答：根据我个人了解，正排索引是资源与关键词的对应，效率较低，倒排索引是关键词与资源的对应，效率较高，solr就是基于倒排索引的，我了解的就是这些。

刁难问法：你们的solr怎么优化的？

回答：我个人的看法就是从这几个方面着手优化：索引打分、索引合并、多core、适当开启cache，最好把solr做成集群，多服务多分片。

13、说说你们的评论是怎么实现的？

这个就是APP打开新闻或者视频详情之后有一个评论功能，输入评论内容点击发送，server端接到请求，调用mongo进行存储，我们存储的内容有用户名、用户ID、评论内容这些信息，同时评论中还包含点赞、回复、举报等功能，如果某条评论被举报，我们的客服人员会通过运营中心审核删除该评论，我们项目当时与mongo做了整合，使用的是MongoTemplate这个门面类来操作。（当点赞、分享、评论完成时，还会获得相应的荔枝树营养值）

刁难问法：你们的回复功能咋实现的？

回答：用户回复我们是存在另外一个collection中，保存的信息有用户名、用户ID、回复内容、评论ID等信息。

14、点赞你们是怎么做的？

点赞我们分为新闻点赞和评论点赞，都差不多，我们都是存到mongo中，这两种点赞是分开存到两个不同collection中，保存的信息也比较简单，就是新闻ID、用户ID或者是评论ID、用户ID。（我们项目当时与mongo做了整合，使用的是MongoTemplate这个门面类来操作）

15、你们的收藏是怎么实现的？

收藏功能较为简单，不涉及复杂的关系，当时我们直接把用户与资源的收藏关系存到了mongodb，同时保存的还有一个类别字段，用于区分不同的资源类型（新闻、视频、直播、电台等）

16、个人中心你们怎么做的？

个人中心就是用户对个人信息的维护，也就包含了上传头像、修改昵称等几个常规功能，没什么介绍的。

17、活动你们是怎么实现的？

我们都是在运营中心上发布活动的，活动发布完之后生成静态化页面，同时把活动信息缓存到redis上，信息包含活动banner地址、静态化页面地址，以json的形式存到redis，APP打开活动广场，server端会从redis取出活动相关信息，返回json给APP。（我们的redis是与spring整合到一块的，使用RedisTemplate这个门面类来操作）

18、工具类和基类怎么封装？

这块其实就是代码的优化，把冗余较多的方法或属性提取封装出来，避免使用硬编码，常用的日期工具类、字符串工具类、json工具类等重用较多的一律提取成工具类，工具类要尽量少的依赖除jdk以外的第三方jar包，基类主要就是bean实例的父类以及一些模板类。主要就是考虑性能、安全、重用率等因素。

19、说说你们项目上线的流程？

20、你是怎么监控项目进度的？

框架概念

1、maven与gradle

maven：

这就是一个项目版本依赖编译打包工具，在开发阶段便于项目快速部署，里面有不同的项目骨架，我们常用的是父子结构多模块开发，常用的命令有：clean、compile、package、install等，为了提高jar包下载速度，我们搭建了nexus私服

gradle：

这是一个与maven差不多的工具，可以理解为是maven的升级版，他依赖于groove语言，抛弃了maven繁杂的xml配置，使用的build.gradle是基于groove语法的，更加容易看懂，逻辑更加清楚，并且支持的仓库有更多的选择性，比如maven仓库、lvy仓库，都有良好的支持，常用的命令有：clean、compile、build等，相比起来，他的灵活性和可扩展性比maven更好。

2、easyui与bootstrap

easyui：
这是一个RIA富客户端框架，开发人员可以使用这个东西做出漂亮美观的页面，而不需要美工，常用的组件有：layout布局、accordion手风琴、tree、tags选项卡、panel面板、datagrid数据表格等，这套框架是基于ajax进行前后台传参，用友丰富强大的实用性，我们的办公平台就是用这个框架搭建的页面，这块是我负责的

bootstrap：
这个是个RIA富客户端框架，栅格系统是他的核心，以响应式布局著称，能够自适应屏幕大小，扁平化风格迎合了大多数用户的心里，他的组件也很丰富，包括炫丽的按钮组、模态对话框、进度条、栅格布局等，是一款不错的页面框架，我们*****这个项目就是用的这个框架

3、jquery与ajax
jquery在我们项目中几乎都有用到，他有丰富的选择器：id选择器、类选择器、标签选择器、父子选择器等，常用的方法有：val()、html()、text()、attr()、each()等，并且还有一些特效方法，诸如：show()、hide()等，使我们对js操作更加方便快捷

ajax是我们常用的异步请求方式，能够实现页面局部刷新，原生态ajax的核心是XHR（XMLHttpRequest），不太常用，我们用的最多的是jquery封装好的ajax，他提供了post、get和ajax不同的请求方式，常用的属性大概有这个几个：url：请求路径，type：请求方式，data：传递的参数，datatype：返回值类型，async：是否同步（默认false），success：成功回掉函数

4、运行原理
struts2：
这个框架我们用的还是比较多的，根据我的理解，首先一个请求发起之后会通过前端拦截器FilterDispatcher进行拦截，匹配到具体的action方法上，执行完相关业务代码返回一个字符串，与action中的result标签对比，跳转到相应页面上

springmvc：
根据我个人的经验来看，这个框架接到一个请求，会被前台拦截器DispatcherServlet拦截，根据RequestMapping把请求映射到具体方法上，执行完相关代码之后，返回视图名称，配置文件中的视图解析器（ViewResolver）会根据返回的视图名称进行视图页面的装配，从而返回对应的页面上

5、springmvc、struts2、springboot
springmvc：
这是个访问控制层，与spring有更好的耦合性，并且是单例，项目中只会创建一个bean实例，基于方法传递参数，避免了线程安全问题，常用的注解有：Controller、RequestMapping、RequestBody、ResponseBody、PathVariable、RequestParam等，页面跳转非常方便，配置一个视图解析器，框架会自动为返回的页面装配前后缀

struts2：
相对来说，这个框架比较古老，他也是一个控制层框架，前后台传参基于模型驱动和属性，相比而言，他的每次请求都会创建一个bean实例，是基于原型模式，对java内存消耗比较大，与spring整合还要借助于一个插件（struts-spring-plugin），比较麻烦

springboot：
这是一款spring微服务框架，只在把开发人员从繁琐的xml文件中解放出来。
他没有各种配置文件，只有一个application.yml（或者application.xml）配置文件，其他的各种框架整合都是在java类中进行装配的。
让开发人员能够更加专注于业务处理，boot框架提供了丰富的注解：SpringBootApplication、Configuration、Bean、RestController、CrossOrigin等，使装配过程更加简单

6、事务及四大特性
事务就是一系列操作为一个单元，要成功都成功，要失败都失败
他有四大特性（ACID）
原子性
事务是不可分割的一个整体，要成功都成功，要失败都失败
一致性
事务操作前后，数据守恒
隔离性
一个事物不能影响另一个事物的执行，相互独立
持久性
事务一旦成功不可逆转

7、spring
这个东西吧也没啥，所有的项目都用到这个框架了，咱们公司估计也一样，他在项目中也就两个作用：第一，创建管理bean实例，我们的congtroller、service、dao都交给了spring管理，说到bean实例，大概说一下bean的生命周期，scope取值有这么几种：singleton（默认）、prototype（原型）、request（每次请求创建一个），session等；第二，控制项目中的事务，利用spring的AOP进行事务拦截，常见的事务传播特性貌似有required、new_required、never等六七种吧，记不太清了，常用的就是required这个；另外，spring还提供了一系列的注解：Service、Repostory、Aotuwired等

8、hibernate、ibatis、mybatis
hibernate：

这是一个orm（对象关系映射）思想的实现，是对jdbc的封装，在项目中充当数据库持久层。
其中有5大河新街口（Configuration、SessionFactory、Session、Query、Trancaction）和两大配置文件（hibernate.cfg.xml、*.hbm.xml），cfg文件中包含数据库信息的配置以及映射文件的路径，hbm文件中包含对象与表的映射关系、主键生成策略、属性与列的映射关系、表与表关联关系
主键生成策略（native、sequence、increment、uuid等），表与表关联关系（一对一、一对多、多对一、多对多等）
访问数据库执行的HQL语句，灵活性差

ibatis：
这个框架也是orm思想的实现，对jdbc轻量封装，手写sql语句，能适应更复杂的业务，灵活、可控性强
同比hibernate，执行效率要高点
dao层需要继承SqlMapClientDaoSupport这个类
映射文件中使用parameterClass接收参数，resultClass返回结果，取参数有两种方式：##和$$，$取值不安全，有sql注入的风险

mybatis：
ibatis的升级版，加入了很多注解，也修改了一些属性
dao层只需要接口不需要实现类，接口中的方法名与映射文件中的id对应
映射文件中使用parameterType接收参数，resultType返回结果，取参数有两种方式：#{}和${}，${}取值不安全，有sql注入的风险
添加了Select、Update、Delete、Insert这些注解，简单的CRUD操作完全能够胜任，方便快捷
另外还有PageHelper分页组件

基础概念

1、String、StringBuffer、StringBuilder区别

String　　字符串常量　不可变　使用字符串拼接时是不同的2个空间

StringBuffer　字符串变量　可变　线程安全　字符串拼接直接在字符串后追加

StringBuilder 字符串变量　可变　非线程安全　字符串拼接直接在字符串后追加

2、List、Set区别

List和Set都是接口，他们都继承于接口Collection,List是一个有序的可重复的集合，而Set的无序的不可重复的集合

List接口实现类有ArrayList,LinkedList,Vector。ArrayList和Vector是基于数组实现的,所以查询的时候速度快，而在进行增加和删除的时候速度较慢，LinkedList是基于链式存储结构，所以在进行查询的时候速度较慢但在进行增加和删除的时候速度较快。又因为Vector是线程安全的，所以他和ArrayList相比而言，查询效率要低

3、HashMap、HashTable区别

HashMap不是线程安全的，HashTable是线程安全。

HashMap允许空（null）的键和值（key），HashTable则不允许。

HashMap性能优于Hashtable。

4、onready和onload的区别

onready比onload先执行

onready是在页面解析完成之后执行，而onload是在页面所有元素加载后执行

onload只执行最后一个而onready可以执行多个。

5、final、finally、finalize

Final：这是一个修饰符

当final修饰一个变量的时候，变量变成一个常量，它不能被二次赋值

当final修饰的变量为静态变量时，必须在声明这个变量的时候给它赋值

当final修饰方法时，该方法不能被重写

当final修饰类时，该类不能被继承

Final不能修饰抽象类，因为抽象类中会有需要子类实现的抽象方法

Final不能修饰接口，因为接口中有需要其实现类来实现的方法

Finally：

Finally只能与try/catch语句结合使用，finally语句块中的语句一定会执行，并且会在return，continue，break关键字之前执行

finalize：

Finalize是一个方法，属于java.lang.Object类，finalize()方法是GC运行机制的一部分，finalize()方法是在GC清理它所从属的对象时被调用的

6、如何在Java中实现线程
继承Thread类或实现Runnable接口

7、Runnable、Thread区别
Runnable是个接口，Java中一个类可以同时实现多个接口，灵活度更高
Thread是个抽象类，由于Java是单继承的特性，继承了Thread就不能继承别的业务类，灵活度差
所以，一般使用实现Runnable接口的方式比较多

8、

线程(Thread)与进程(Process)

进程定义的是应用程序与应用程序之间的边界，通常来说一个进程就代表一个与之对应的应用程序。

一个进程可以包括若干个线程，同时创建多个线程来完成某项任务，便是多线程。

9、线程的几个状态
新建：new完之后就是新建
就绪：调用完start方法，随时可以获取cpu资源
运行：获取到cpu资源，开始运行
阻塞：在运行时由于某些原因放弃cpu的使用权，但是又没有执行完
死亡：执行完了，或者说异常退出了run方法

10、forward、redirect

从数据共享上：

Forword是一个请求的延续，可以共享request的数据

Redirect开启一个新的请求，不可以共享request的数据

从地址栏：

Forword转发地址栏不发生变化

Redirect转发地址栏发生变化

11、servlet生命周期
实例化
初始化
提供服务
销毁
不可用

12、jsp九大隐式对象、四大作用域
隐式对象：

输入/输出对象： request response out

作用域通信对象：session application pageContext

Servlet 对象：page　config

错误对象：exception

作用域：

四个作用域从大到小：appliaction>session>request>page

application：全局作用范围。

session：会话作用域。

request：请求作用域。

page：一个JSP页面。

13、json与xml
这两个都是数据传输的载体，都能够跨平台传输数据
xml：
复杂度较高，老项目用的比较多
json：
清晰易懂，现在比较流行

14、dom4j与sax

dom4j不适合大文件的解析，因为它是一下子将文件加载到内存中，所以有可能出现内存溢出，

sax是基于事件来对xml进行解析的，所以他可以解析大文件的xml

所以dom4j可以对xml进行灵活的增删改查和导航，而sax没有这么强的灵活性

所以sax经常是用来解析大型xml文件，而要对xml文件进行一些灵活（crud）操作就用dom4j

15、冒泡排序
整体来讲，冒泡比较简单，核心思想就是这样
双层循环
比较大小
交换位置

16、描述static

static变量

按照是否静态的对类成员变量进行分类可分两种：一种是被static修饰的变量，叫静态变量或类变量；另一种是没有被static修饰的变量，叫实例变量。

static方法

静态方法可以直接通过类名调用，任何的实例也都可以调用，因此静态方法中不能用this和super关键字

static代码块

static代码块也叫静态代码块，是在类中独立于类成员的static语句块，可以有多个，位置可以随便放，它不在任何的方法体内，JVM加载类时会执行这些静态的代码块，如果static代码块有多个，JVM将按照它们在类中出现的先后顺序依次执行它们，每个代码块只会被执行一次

static和final一块用表示什么

static final用来修饰成员变量和成员方法，可简单理解为“全局常量”！

17、你对java虚拟机（jvm）的理解
我自己看过Java虚拟机的规范，多少了解一点

在 Java虚拟机规范中，一个虚拟机实例的行为是分别按照子系统、内存区、数据类型和指令来描述的，这些组成部分一起展示了抽象的虚拟机的内部体系结构

方法区
存放了方法信息、字段信息、常量池等

堆
这个里边存储的是各种实例对象以及数组对象等，并且Java虚拟机只有一个堆空间，所以所有线程都会共享这个堆

Java栈
这个里边存的都是各个变量名之类的东西，好像叫指针，

包括局部变量、参数、返回值等

业务场景

1、简单说一下文件上传的原理（上传文件步骤）
我们在做项目的时候经常会遇到文件上传的问题，比如我之前做那个*****项目，更换头像就是文件上传
实现的步骤很简单，就这么几步
第一呢，页面必须使用form表单，以post方式进行提交
第二，更改表单中enctype的属性值为multipart/form-data
第三，表单中必须有一个文件域
第四，后台使用File类接收文件并且使用IO流的方式写到硬盘上
最后把文件存储的路径保存到数据库中
整个过程需要注意的是资源的释放以及释放顺序，我们项目中直接被我封装成了文件工具类，调用就可以
这种方式，文件是随着表单一起提交的，另外，我们还使用uploadify做过文件异步上传的功能

2、如何避免表单重复提交
重复提交表单是很恶心的一件事，所以我们项目中也做了处理
我个人对避免重复提交表单有几点看法
第一呢，可以使用loading加载项，点击完提交，页面开始转圈圈，直到成功跳转页面
第二种方式呢，点击完提交按钮就把提交按钮置为不可用状态，也能避免，我们项目中使用的就是这种
第三呢，让后台处理，用户跳转到添加页面的时候，后台生成一个uuid保存在session对象中，前台取出uuid放在隐藏域中，当用户点击提交的时候，后台先取出session中的uuid，与表单提交的uuid对比，如果一样，移除session中的uuid并插入这条数据，否则不做任何操作直接返回页面

3、分页如何实现（分页步骤）
分页这个东西每个项目都会涉及到，我们大部分的项目都是使用的第三方分页组件，自己也封装过，比较麻烦，当时是这么做的
第一，页面封装一个显示分页的组件
第二，后台封装一个分页的工具类
第三，查询出总条数，使用工具类计算出总页数、开始位置、结束位置
第四使用分页sql语句根据计算结果查询列表
第五，前台相关页面引入分页组件
需要注意的是：查询总条数的where条件要和查询列表的where条件保证一致

4、junit（postman、soapui）
这个东西我们经常用了，junit就是为java量身打造的测试工具
每当我们做完一个功能的时候，如果启动项目来测试，太慢，修改代码还要重启，所以我们直接使用junit测试
由于项目都是用了spring框架，所以我们把junit与spring做了整合（加一个test-spring的jar包）
常用的注解有：RunWith、ContextConfiguration、Test、Ignore等

另外，我还使用过postman和soapui这两个测试工具
我们之前做的app项目就是用postman来测试接口的，他能发送各种各种请求（post、put、delete等），能传递不同类型的参数（form、xml、json等），能与各浏览器继承，使用方便

soapui主要我们拿来测试发布的webservice接口，比较实用

5、系统管理（权限管理）
我在*****项目中负责的就是这块，我们当时是这么做的，设计出5张数据表（用户表、角色表、权限表、用户角色关联表、角色权限关联表）
整个流程是这样的，当一个用户登录时，验证用户密码是否合法，通过验证之后，我们会根据用户id到用户角色关联表中查出关联角色的id集合，然后根据角色id到角色权限关联表中查到关联权限的id集合，之后拿着查到的权限id集合到权限表查询出具体的权限列表，将权限列表保存一份在redis缓存上，然后把权限列表展现给当前登录用户，从而达到不同用户看到不同权限列表
同时呢，为了防止用户跨权限（输入url）来访问，我们在项目中添加了权限拦截器（PermissionInterceptor）来验证

还有就是，为了把权限控制的更加严格，我们把权限分为菜单级别和按钮级别，从而实现控制权限到按钮

6、git
我们最近做的这个*****项目就是使用的git作为版本控制器，操作也比较简单
首先与开发工具集成起来，关联上代码托管中心，把代码库中已存在的项目clone下来
之后在本地创建一个分支branch，在分支上完成自己负责功能的开发，自己测试没问题之后，提交代码到自己的分支上，然后合并自己的分支到主分支上，使用pull命令拉取版本库上的最新资源，然后push本地修改到远程版本库上
我们之前代码托管在github上，需要生成并配置公钥私钥才能关联

7、redis

redis是一个基于key-value的支持多种数据类型的可进行持久化的内存数据库。

我们在项目中使用的时候为了保证redis服务器的安全，通常会在redis.conf配置文件中绑定具体的ip地址，这样只有该ip地址才能访问redis服务器，并且设置长度为20位左右的密码，从而保证只有进行了密码授权才能进行相关的操作。

为了保证redis不会因为占用内存过大而导致系统宕机，通常在将redis当做缓存服务器使用时，设置存储数据的过期时间，并且通过设置maxmemory【最大内存】和maxmemory-policy【数据清除策略】为allkeys-lru来达到预期的效果。

我们在项目中通常使用redis来充当缓存服务器来缓存用户权限列表、首页公告信息、轮播图信息等

使用了jedis作为客户端，并考虑到性能问题使用了jedis连接池。

考虑到redis服务器的高可用性，我们做了redis的主从复制，并且通过加入哨兵来使redis主服务器宕机时，从服务器自动转换为主服务器继续提供服务

8、mongodb
mongodb是一个非关系型数据库，弱事务，以文档（document）的形式进行数据存储，存储的是bson（json）格式的数据，我在*****这个项目中用到了这个
当时我负责了评论管理（新闻的评论），考虑到新闻数据量比较大，评论的数量可能成几何倍数增长，oracle/mysql存那么多数据，查询效率肯定跟不上，所以我采用了mongo来存
评论包含新闻ID，用户昵称、评论内容等信息，由于页面是静态化的，评论的提交我们采用的ajax跨域的方式，后台采用mongoclient来连接mongo进行操作，考虑到后期数据量会很大，所以我们配置了mongo分片，把数据分片存到不同mongo服务上，每个分片都只有一个单节点，对于容灾有点差，所以每个分片上我们配置了副本集，整体上就是这样
另外我们还有新闻点赞也是这么实现的
除此之外，我们还使用springAOP的特性做了前置通知进行用户行为的统计

9、定时器
这个东西不是我做的（微笑），这是我带我们公司那个实习生做的，业务也比较简单，就是在各个法定假日向我们的客户发送节日慰问短信，整体上逻辑不算复杂，为了定时任务更加灵活，我让让使用的quartz定时器，能够通过cron表达式设定不同时间点的定时任务，但是配置文件有点小复杂，要配置调度中心，任务类，任务详情等好几个bean，多个任务还要多次配置，麻烦，我让他使用的注解方式来完成，一个注解搞定一个任务（Scheduled），方便快捷
至于发送慰问短信，我们公司购买了阿里云的短信包，融入SDK，按照api调用就好

10、找回密码（发送短信验证码）
这个功能是我负责的，为用户提供一个密码找回的途径，我是这么做的：
首先用户输入手机号，然后异步校验手机号是否存在，不存在提示重新输入
然后点击发送手机验证码，同时把六位验证码存入数据表中，（页面把发送验证码按钮置为不可用，同时倒计时60秒，60秒之后按钮置为可用，并且显示重新发送）
用户输入手机验证码和新密码，点击找回按钮
后台校验验证码是否过期（15分钟），没有过期则比对验证码是否正确，正确，将密码进行重置，同时删除修改表中的验证码为不可用（一条验证码只能使用一次）
重置成功，页面跳转到登录页面
嗯整体流程大概就是这样

11、报表（highcharts/echarts）
报表在这几年接触还是挺多的，能更加直观的看到各项数据指标的走势，之前我做的这个*****项目里，我就负责的报表中心
当时我们是要按季度、按月来统计处用户注册量、会员注册量、员工出勤率等报表，并且支持数据下钻，当时我们采用的highcharts来实现的报表，整个功能的核心，在我看来就是sql语句这块，只要sql写对了，能查出对应格式的数据，生成图表就毫无压力了

12、httpclient
这是一个模拟http协议的接口调用技术，是近几年比较流行的一个接口调用的实现，他能模拟出不同请求方式（put、delete、get、post等），支持json传递，表单传递，xml传递，使用起来比webservice简单
我之前做的这个项目就有用到这个技术，我们公司有自己的用户中心，用户信息统一管理，提供了登录、注册、修改用户信息等几个功能，于是我们项目的用户模块直接就是调用用户中心提供的接口，使用的就是httpclient，我是这么实现的

根据接口文档封装好本地的基类
然后根据url路径创建出对应请求方式的对象（GetMethod、postMethod等）
封装请求参数（RequestEntry）
执行execute方法，获取返回结果
嗯基本上就是这样

13、webservice
这是一个跨平台跨语言的项目通讯技术，能够把不能项目的数据实现共享，在我们项目中有这么一个功能，需要获取到用户手机的归属地，自己实现这个功能不太现实，数据采集就是一大难题，于是我们就是调用的webservice接口，当时我们使用的是cxf框架，首先使用wsdl2java命令根据wsdl地址生成本地类，放到项目中交给spring管理，注入到控制层，然后当做本地类一样调用

14、freemarker
页面静态化用的还是比较多的，我们做的*****项目，我负责的就是新闻管理，由于新闻查询量比较大，几乎没有修改操作，所以，我为新闻生成静态化页面，部署到另外一台服务器上，这样能够减缓项目服务器的压力，页面静态化的核心就是：data + templates = page
我的实现流程是这样的：
首先定义页面模板，查询出新闻信息，与模板一块生成页面，由于初始化时数据量大，所以采用分页的方式批量生成页面（此处使用到了递归），生成完页面的数据，我修改了新闻状态为已被静态化，防止重复生成页面，为了防止文件名重复，我采用的新闻ID作为文件名，当新闻有变动时，可以直接覆盖或删除
为了保证新闻的及时性，我并没有采取定时器来生成页面，当一条新闻录入数据库时，紧接着我就调用了查询并且马上生成静态化页面。
整体情况就是这样

15、nginx+tomcat

我们在做*****这个项目时，考虑到服务器性能的问题，最开始想到使用纵向扩展，来增加硬件的配置提高其性能，但这样做比较耗费资金，而且服务器内存空间也是有限的。所以我们采用横向扩展来实现

当时我们使用nginx+3个tomcat进行负载均衡，在我们不进行负载均衡之前，那所有的请求都由一台tomcat进行处理，这样会使我们的tomcat所承受的压力增大，而我们进行负载均衡之后，同样数量的请求经过nginx将其分发到多台tomcat进行处理，从而降低每台tomcat所承受的压力，而且当其中一台机器宕机时，其他机器还可以继续提供服务，保证服务不间断。

当时项目在部署完成后，遇到这么个问题，用户登录输入验证码的时候，明明验证码输入的正确，但总是提醒说验证码不正确从而不能正常登录，经过分析后发现有可能第一次请求被发送到t1上，那么放在session中的验证码就被放到了t1上，当用户输入验证码点击登录时，新发送的请求有可能被发送到t2上面，这样在进行对比时就肯定会不一致从而提示验证码输入错误，后来我就考虑使用ip_hash这种负载均衡策略来代替默认的轮询策略，虽然解决了验证码错误问题，但是在后续的测试中发现如果用户在使用过程中突然一台服务器宕机了，那么因为session在这台服务器上存储着，所以就会提示用户重新登录，这样使用户的体验度非常不好，最后就通过将session信息保存到redis服务器中从而在多台web服务器中实现session共享，这样就解决了上面所说的那些问题。

（这一点不问不用说）：为了解决单点故障

同时我们为了避免ngnix的单点故障，就是在ngnix前面加上一个f5，分给多个ngnix，再通过ngnix分发给多个tomcat

16、dubbo+zookeeper
这个是一个RPC远程服务的实现，旨在更加快捷稳定的进行项目通讯，并且可以量化部署，集群式提供服务，容灾容错较强
在我们****这个项目中，考虑到用户量比较大，后台的负载量也比较大，所以我们做了dubbo集群部署，把项目分为控制层和业务层两个单独的项目，业务层就是服务提供者，使用dubbo把服务注册到zookeeper注册中心上，控制层是服务消费者，通过dubbo到zookeeper注册中心获取服务，dubbo支持各种协议：http、dubbo、hession等，灵活度高，为了防止参数泄露，我们采取了隐式传参的方式进行参数传递，另外，dubbo是基于长连接的框架，不会频繁连接与断开，节省效率，基本上就是这样

17、ftp
这是一个文件服务器，我们很多项目中都用到过，就拿*****这个项目来说，我们有一个资源管理中心，主要是把各种资源（视频、文档、图片等）分门别类进行归档，项目服务器硬盘有限，不能存储太多的东西，所以我们搭建了ftp服务器，当时我们用FZServer来搭建的，后台操作是这样的，把用户上传的各类资源通过FTPClient客户端，以字节流的方式写出到ftp服务器上，为了防止文件重名覆盖，我们使用的uuid生成文件名，为了防止文件被破坏，必须设置Binary为true

18、单终端登录（这头登录那头掉线，那头登陆这头掉线）
在我们的*****项目中，为了控制同一个账号同时只能在一台终端设备登录，我们做了这个功能
实现的思路大概是这样的：
由于我们把每个账号的登录信息都存在当前会话的session中，所以项目启动的时候我们初始化一个Map来作为session池，把每一个用户的session都保存在里面，以用户ID作为键，session对象作为值
每当用户登录成功，我们会根据用户ID判断这个session是否存在，如果不存在则保存，如果存在则判断sessionID是否一致，如果一致不做任何操作，如果不一致，那就先让原来的session失效，在保存当前的session
我们就是通过这样方式实现的这个功能

19、poi
我们的通讯录管理中有一个导出功能，按照当前搜索条件导出所有检索到的通讯录信息，考虑到以后数据量过大，我们采用的是分页的方式导出
导出数据的流程是：
首先查出当前页结果
创建出excel对象（HSSFBook）
为excel创建工作簿（sheet）
再为工作簿创建行（row）
每一行创建单元格（cell）
把信息填充到单元格中
判断是否是最后一页，如果不是最后一页，递归调用自身继续查询，最后将excel文件已下载的方式写出到浏览器端来完成导出功能

另外，我们的poi还应用于导出每季度、每月、每周的报表信息

20、solr
为了提高我们项目中的检索效率，我们采用了第三方索引，之所以没有使用数据库中索引，因为动态维护的时候，增删改时候，耗费时间效率，所以我们选择了 第三方索引。
当时考虑的是Solr和Lucene：
首先，Solr与Lucene 并不是竞争对立关系，恰恰相反Solr 依存于Lucene，因为Solr底层的核心技术是使用Lucene 来实现的，
那么，Solr和Lucene的本质区别大概就是：Lucene本质上是搜索库，不是独立的应用程序，而Solr是。
Lucene专注于搜索底层的建设，而Solr专注于企业应用。Lucene不提供搜索服务所必须的管理，而Solr提供，而且附带了一个基于HTTP 的管理界面。
所以，我们最后决定使用Solr；

Solr服务器是由我负责搭建的：
服务器的搭建我用了一周的时间，包括：与数据库的交互，调用者的交互，以及一些业务上的设定，比如：搜索显示高亮，中文分词器（这里我用的是IK分词器），拼写检查，检索建议（autocomplete），分组统计（solr的Facet组件），相似匹配，拼音检索优化时间的选定等等
项目前期使用了一台服务器，后来并发量大了发现一台tomcat承受不了，所以就选择负载均衡-增加服务器
负载均衡的话，当时也是在zookeeper与nginx之间选了一下，由于只准备增加2台服务器（节点），nginx使用起来比较简单顺手所以就采用了nginx。

数据库篇

1、代码优化
主要考虑两个方面：可维护性和可读性
1、编码规范（注释）
2、工具类和基类的封装
3、避免syso输出（使用日志打印）
4、避免String拼接字符串（StringBuffer）
5、IO流用完及时释放资源
6、尽量使用软编码

2、业务优化：
主要目的提高用户体验度，所以我们会进行一些列的优化，如：
1、使用回车提交表单或查询数据
2、查询结果最好按照一定规则排序返回
3、友情提示（尤其是删除操作）
4、进度条（避免用户重复提交）

3、sql优化
工作这几年，写过不少的sql语句，sql语句的好坏直接影响效率，所以我们对sql会有一些优化：
1、尽量把sql语句都写成大写，oracle会把小写转成大写执行
2、避免在索引列上使用计算和函数，会导致索引失效
3、数据量过大时避免like查询（lucene、solr）
4、避免使用*查询
5、避免使用>、<关系符，替换为（>=、<=）

4、你对osi网路七层架构的理解
网络数据传输的途径，越是靠近底层框架的，速度越快，越不安全，越是靠近上层框架的，速度越慢、越安全

不再总结，自己看宝典就行

人事问题

个人相关

1、你的三年职业规划是什么
2、你怎么看待频繁跳槽
3、你的优点和缺点是什么
4、看过哪些技术书籍
5、时针走一圈，时针分针重合几次
6、讲一件你印象最深的事情
7、与上级意见不一致的时候，你会怎么办
8、你在工作中遇到的最大的困难是什么
9、说一件能证明你能力的事情
10、你为什么值15k
11、你平时有什么爱好
12、你对一份工作更看重哪些方面？平台，技术，氛围，城市，money
13、如果你被录用，过一段时间你发现不适合这个职位，你怎么办
14、你的薪资有哪几部分组成
15、你为什么要离职（离职原因）
16、你住哪里
17、平时上班需要多长时间
18、有没有缴纳五险一金
19、你属什么的
20、你今年多大了
21、你都被派到过哪家公司上班
22、你工作3年一直在这家公司吗
23、你什么时候离职的

公司相关

1、你们公司规模多大
2、你们公司地址在哪
3、你们公司有几个部门
4、你们公司有多少个开发团队
5、你们团队有多少人
6、你们公司都从事什么业务

学校相关

1、你什么时候入得学啊
2、你学的是什么专业
3、你们学校有多少院系
4、你大学都学过什么课程
5、你哪年毕业的
6、你们学校的地址是哪里
7、你们学校在市区还是郊区
8、大学时，你们班多少人
9、从你们学校到火车站怎么走
10、你们系主任叫什么名字
11、你们校长叫什么名字
12、大学谈过恋爱吗
13、为什么跑那么远的地方上大学