**自我介绍**

我叫陈鑫晖。从事java后台开发三年。此次来公司应聘java工程师职位。

在此之前，做过的项目主要是 在线教育、上家公司百家互联，主攻线上教育的一个科技公司，最近做的是一个微服务的项目，主要涉及的架构为springBoot+SpringCloud，

技术方面 用户登录注册采用 jwt 加密验证token令牌的方式

代码整体使用springdatajpa

注册短信使用阿里云通信

利用es分布式搜索引擎进行搜索+ik进行关键字具体搜索........等等

我都参与过并且熟练掌握，

我认为 个性比较沉稳，沟通能力较强。在工作上能够认真负责，喜欢不断的去挑战自我，突破自我，对新事物有较强的接受力。

**上家公司大致信息**

公司名称：北京百家互联科技有限公司

公司地址：北京市海淀区西北旺东路10号院东区7号楼1层W102

学校、家庭住址。

上家公司薪资 入职12K 离职 15K

学校 北京东方研修学院 15年7月毕业

离职原因 合同到期签的两年的合同，2019年3月到期，这两个月主要在公司做收尾和考虑续签合同。 因为想提高自己接触更高的层次，所以最后选择换个不同的环境成长。

**对上家公司的评价**

很感谢上家公司,在上家公司学到了很多技术,在业务逻辑方面得到很大的提升,积攒了宝贵的经验,也得到很多同事和领导帮助,对我的成长很有帮助;

**项目开发模式**

软件一般分为：瀑布式开发和迭代式开发、敏捷开发

**瀑布式开发**：是最典型的预见性的方法，严格遵循预先计划的需求分析、设计、编码、集成、测试、维护的步骤顺序进行。

瀑布式的主要的问题是它的严格分级导致的自由度降低，项目早期即做出承诺导致对后期需求的变化难以调整，代价高昂。

瀑布式方法在需求不明并且在项目进行过程中可能变化的情况下基本是不可行的。

**迭代式开发**：每次只设计和实现这个产品的一部分

逐步逐步完成的方法叫迭代开发

每次 设计和实现 一个阶段 叫做 一个迭代

每一次迭代都包括了需求分析、设计、实现与测试。

迭代式开发的优点：降低风险，持续的测试和集成

**敏捷开发：**是一种应对快速变化的需求的一种软件开发能力，更强调程序员团队与业务专家之间的紧密协作、面对面的沟通（认为比书面的文档更有效）、频繁交付新的软件版本、紧凑而自我组织型的团队、能够很好地适应需求变化的代码编写和团队组织方法，也更注重软件开发中人的作用

**项目开发流程**

**项目一般分为：**

1. **立项阶段：（合同，建设团队）**
2. **需求调研阶段：（需求文档、需求规格说明书）**

需求规格说明书是开发团队把从客户那里理解的需求，以书面规范的方式形成的说明文档，目的是使用户和开发者对软件的初始规定有一个共同的理解。

内容一般包括：

所有文档公共部分：（目的、背景、术语定义、参考资料、预期读者）

总体设计：功能需求、性能需求、总体设计概念、总体技术框架、基本处理流程等

1. **设计阶段：（概设、详设、demo）**

概设：

在需求规格说明书基础上，对系统进行进一步的详细设计：主要包括：系统功能架构、系统逻辑架构、系统功能模块设计、功能需求追溯、接口设计、数据结构设计等；

详设：

对系统各个层面进行详细的规划和设计，包括:界面要求、开发规范、用例图、约束、详细数据库表设计、功能开发流程描述等。

1. **编码阶段：（开发计划、开发进度、工作日报、周报）**
2. **测试阶段：（测试用例、测试文档、操作手册）**

测试用例：开发没有完成之前，测试人员就要根据需求和详细设计，预测各个模块的运行流程和预期效果，并书写成测试用例；

1. **试运行阶段：（内部验收文档、部署实施文档）**
2. **正式上线阶段；**
3. **运维阶段：（维护文档，客户验收文档）**
4. **完结阶段；**

**跟谁学现在的定位是B2C模式的在线教育公司。**

**1.人员配备**产品经理：1人，负责需求以及给出产品的原型图  
项目经理：1人，负责项目管理以及模块分配  
前端团队：2人，根据产品经理给出的原型制作出静态页面  
后端开发团队：4人，分为两个小组，每小组2人，为了更好的实现产品功能  
测试团队：2人，测试所有的产品功能  
运维团队：2人，项目的发布以及维护。  
15人

**项目整体介绍**

项目整体采用SpringBoot、SpringCloud的微服务架构，数据层springDataJPA， 所用环境都是基于Docker环境，测试工具用的是Postman（提供功能强大的Web API & HTTP 请求调试），数据库在生产环境中要分片部署(MyCat)通过Eureka实现注册信息集中化管理，微服务之间通过feign相互调用。在微服务架构中通常会有多个服务层调用,然后就用到了熔断器(Hystrix),因微服务一般有不同的网络地址，用到了微服务网关(zull),由于服务数量巨多，为了方便服务配置文件统一管理，实时更新，所以用到了分布式配置中心组件(SpringCloudConfig),最终那我们把配置文件传到了码云上，为了不重启服务,节省时间,就用到了(SpringCloudBus)。我们还运用了(Jenkins)实现持续集成,实施监控集成中存在的错误，提供详细的日志文件和提醒功能，还能用图表的形式形象地展示项目构建的趋势和稳定性。短信验证方面用的是阿里大于短（阿里云通信）信接口，使用消息中间件RabbitMQ的消息队列通知短信微服务调用阿里短信接口下发短信，使用Redis存储防止用户多次恶意发送短信。密码加密用的是BCrypt， Spring Security 提供了BCryptPasswordEncoder类,实现Spring的PasswordEncoder接口使用BCrypt强哈希方法来加密密码。

**项目模块详细介绍**

1. **搜索模块全文检索**：搜索模块使用的是elasticsearch来完成，个人感觉es的使用还是比较简单的，我们使用spring-data整合es完成课程的发布与浏览操作。在使用过程中我们在处理搜索关键字解析时用到了ik分词器，默认的中文分词是将每个字看成一个词，这显然是不符合要求的，所以我们需要安装中文分词器来解决这个问题。IK提供了两个分词算法ik\_smart 和 ik\_max\_word其中 ik\_smart 为最少切分，ik\_max\_word为最细粒度划分,我们在项目中使用的是ik\_max\_word模式。

ES的核心搜索引擎是通过倒排索引实现的，说倒排索引前我先说一下正排索引，它是通过关键词去文档中直接查找相应的单词，而倒排索引则不同，它是先把文档中的内容切分成词条，再经过词干还原、不区分大小写等方式转变成词项存储在倒排索引的数据结构中。倒排索引的数据结构包括由单词组成的词典、单词索引以及关键词出现过的文档编号。当用户在文本框中输入关键词后，es搜索引擎会先去词典中进行查找，找到后会根据单词索引展示文档内容。

**IK分词器**

1. ik\_max\_word会将文本做最细粒度的拆分,比如会将”中华人名共和国人民大会堂”拆分为”中华人民共和国\中华人民\中华\华人\人民共和国\人民\共和国\大会堂\大会\会堂”等穷尽各种可能的组合.
2. Ik\_smart会将文本做最粗粒度的拆分,例如[中华人民共和国国歌]会被拆分为[中华人民共和国\国歌]
3. 注册登录模块：在注册模块，在用户点击发送验证码通过rabbitmq向消息微服务发送一个map类的消息，这个消息包含手机号、验证码等信息（验证码在用户模块已经提前保存在redis中key为sms+手机号，value为验证码），这样消息微服务会对该队列进行监听，接收到消息后调用阿里大于短信接口发送短信，完成注册业务。为了防止有恶意用户进行对接口的调用，前台可以采用把发送验证码的按钮倒计时置灰，后台采用的是布隆过滤器来实现对恶意用户的拦截，如果用户第一次获取验证码，则将手机号经过多个Hash算法为一个整数key，映射到该全为0的bit数组，把key对应的位置的值0变为1。如果过滤器过滤到该手机号对应的位置值为1，则说明该手机号已经获取过验证码（恶意用户）。其中我们用到了redis来统计手机号注册次数并判断执行锁定一小时或者永久冻结等相关操作。

**用户进行登录的时**候，去访问jwt的接口，登陆成功以后通过jwt会给用户返回一个token的值，我们主要是把用户的信息保存在Threadlocal中，我们会把userid查出来通过jwt进行封装返回，当通过前台业务把jwt返还回来之后把userid再取出来并且将它放入Threadlocal中，它是一个线程绑定的存储空间，相对安全。注册密码使用MD5Util的工具类的encrypt的方法将用户密码传入进去加密，我们登陆首先会从数据库当中获取信息，获取到的结果与加密之后的密码进行匹配，匹配成功就可以完成登录。

**支付模块**

**支付宝：**当用户点击提交订单的时候,会跳转到支付页面,在支付页面生成支付宝的二维码,并显示订单号和金额,用户用手机扫描二维码,完成支付功能.当用户支付成功以后跳转到支付成功页面,支付失败的话跳转到错误页面

这块我们项目经理利用公司的一些资质申请了支付宝的企业账号，主要是为了获取支付宝应用ID(APPID)、网关、以及利用支付宝rsa秘钥生成器生成公钥和私钥。

**开发步骤：**

1、首先拿公司资质去支付宝申请商铺号和秘钥，成功后支付宝会给你返回一个API，里面有商铺号、秘钥、支付的URL、回调的URL格式、传递参数的规格等。

2、点击“支付”，会跳进自己写的一个controller方法中，从页面传过去的有订单号和价格总和，再把商铺号、秘钥、回调的URL一起add到一个list里。

3、用httpclient把list根据支付宝给的支付URL传递给支付宝。

4、此时支付宝会把你传递过来的商铺号和秘钥进行比对，如果成功，支付宝会以html的方式返回一个选择银行支付的界面。你进行支付就OK。

5、你支付完成后，支付宝会把流水号、订单号、价格返回，此时你controller的回调方法里会把这些数据进行比对，都正确后给支付宝发送一个“SUCCESS”，支付宝就知道支付已经成功，页面就跳转到支付成功页面。反之就跳转到支付失败页面（这里所有支付宝返回给我们的页面都是支付宝自己的页面，我们不需要写）

6、最后把流水号等相关数据拿去数据库进行一个操作。我们是订单表和购物车表为一张表，此时只需要根据订单号把订单状态从0（购物车）改为1（付款后的订单），然后再把流水号存入。此时整个支付就完成了。

流水号主要负责的是商城数据库的价格与第三方支付方价格进行对比！！成功则审核通过并发货！

****评论回答模块**在我们的这个线上直播授课中，学员的问答和评论模块的数据量相对较大，而且写入操作频繁，对数据库会造成很大的压力，并且数据价值相对来说比较低。所以我们使用了MongoDB来进行存储。使用MongoDB的好处呢，就是读写快速，他将热数据存储在物理内存中，使的热数据的读写变得十分快速，提高了整体的速度和效率，而且它使用bson的存储格式，十分适合文档格式的存储于查询。这些优点正好与我们的业务相契合。**

**我们现将提问评论所需要的表字段，发布人Id，吐槽内容，点赞数，上级ID等封装起来，使用spring-data提供的MongoDBTemplate工具类，调用insert方法将吐槽数据存入MongoDB中，这样我们就可以直接从MongoDB中进行查询了，再比如说，我们的吐槽点赞，我们只需要将点赞数加1就可以了，没必要从数据库中查询出所有的字段后在更新所有字段，这个时候我们就可以用MongoDB只修改其中的点赞数的字段。评论和吐槽其实是差不多的实现思路，我们用MongoDB来存储这些数据量大、读写频繁、价值低的数据从而减轻我们数据库的压力，实现业务的优化。**