# 自我介绍

你是谁？

来自哪里？

哪上的学？ 什么时候毕的业？

怎么从事java这个行业的?

都在哪工作过?

公司是干嘛的?

在这些公司你做了哪些项目?

最近的一个项目稍微详细一些?

面试官，你好！我叫李鑫，老家呢是山西的，今年25了。在山西工商专修学院读的大学，15年毕业的。

在大学呢我学习的是计算机专业,(计算机科学与技术),所以毕业以后呢,我进入到了 药渡经纬信息科技有限公司 .从事了java开发的这个行业

(毕业学校: 山西工商专修学院

学校地址: 山西省太原市小店区坞城南路99号 俩校区(北格校区[会计和建工两个专业],龙城校区)山西财经大学华商学院在学校北面

学院名称: 山西太原师范学院

学院院系: 计算机科学与技术

入学毕业时间: 入学时间 2011.9.21 毕业时间 2015.7.20

学制年限: 4年

学历证书: 本科

大学课程: C语言程序设计、电子技术、数据结构、操作系统、数据库原理与应用、计算机组成原理、计算机网络、Java开发、大数据处理与分析、三维建模技术、嵌入式应用与开发等)

经营范围：技术开发、技术转让、技术咨询、技术服务、技术推广；计算机技术培训；基础软件服务；应用软件服务；计算机系统服务；数据处理；销售计算机、软件及辅助设备、电子产品、机械设备、通讯设备。（企业依法自主选择经营项目，开展经营活动；依法须经批准的项目，经相关部门批准后依批准的内容开展经营活动；不得从事本市产业政策禁止和限制类项目的经营活动。） 在公司里呢,我从入行以来呢,先做了一个小型的门户网站,然后就是OA系统,第二年做了一些B2C的电商项目,和教育项目

(

北京市海淀区上地信息路7号5层508

乘车：地铁西二旗转一站公交 上地七街下车即到

联系电话：86-010-82157179) 15269436992 \*\* 项目经理

13167322093 \*\*\* 项目经理  
18146570512

1833013442 \*\*\* 人力资源

我最近参与的项目是 医学慕课

**面试官您好，我叫李鑫，山西人，是来面试咱们Java开发工程师的，15年毕业后，一直在做java开发， 我在工作期间做过五个项目，在这期间用到的框架有Spring+SpringMvc+Mybatis这些主流的框架, 数据库方面有mysql关系型数据库及redis非关系型数据库，还用到一些技术，例如solr，CAS、等都接触过，通过这几年的工作，对于自身的能力也有很大的提高。最近做的就是一个医疗教育视频学习平台的项目。**

**【开发工具】:Idea+Tomcat+JDK1.8+Maven+MongDB+Redis**

**【开发环境】:SpringBoot+SpringCloud+SpringMVC+SpringData+ActiveMQ**

**【责任描述】负责登录，注册模块、负责搜索模块、负责上传课程模块、负责会员充值模块、负责个人信息模块**

**【项目描述】**该项目是一个面向医学人群的医学视频教育平台。有手机APP端和PC端，分为前台用户系统和后台管理员管理系统。前台用户系统主要是用户浏览、搜索相关视频、观看视频学习的；后台管理员系统主要是管理员审核上传视频、管理用户、发布公告、医学新闻等。用户前台分为搜索模块、首页、公开课、时间提升、病例库、学术课件、上传视频等模块；轮播图，用户点击不同图片可跳转相关内容；科室分类模块，分为内科、外科两大类，不同类别中又细分为不同小类别，用户通过点击不同类别可搜索到相关类别的视频内容。平台还有登录模块与会员充值模块。

登录、注册

精简版：

##### JWT的结构

JSON Web Token(JWT)是一个开放标准(RFC 7519)，它定义了一种紧凑的、自包含的方式，用于作为JSON对象在各方之间安全地传输信息。该信息可以被验证和信任，因为它是数字签名的.

JSON Web Token由三部分组成，它们之间用圆点(.)连接。这三部分分别是：Header(头部),Payload(载荷),Signature(签证).

header典型的由两部分组成：token的类型（“JWT”）和算法名称（比如：HMAC SHA256或者RSA等等）。然后，用Base64对这个JSON编码就得到JWT的第一部分

JWT的第二部分是payload，它包含声明（要求）。声明是关于实体(通常是用户)和其他数据的声明。声明有三种类型: registered, public 和 private.对payload进行Base64编码就得到JWT的第二部分.注意，不要在JWT的payload或header中放置敏感信息，除非它们是加密的。

签证是header与载荷都经过BASE64位加密后形成的字符串与秘钥经过header中申明的算法一起加密形成签证。签名是用于验证消息在传递过程中有没有被更改，并且，对于使用私钥签名的token，它还可以验证JWT的发送方是否为它所称的发送方。

##### JWT的 6优势

1、支持跨域访问: Cookie是不允许垮域访问的，这一点对Token机制是不存在的，前提是传输的用户认证信息通过HTTP头传输.

2、无状态(也称：服务端可扩展行):Token机制在服务端不需要存储session信息，因为Token 自身包含了所有登录用户的信息，只需要在客户端的cookie或本地介质存储状态信息.

3、 去耦: 不需要绑定到一个特定的身份验证方案。Token可以在任何地方生成，只要在你的API被调用的时候，你可以进行Token生成调用即可.

1. 性能: 一次网络往返时间（通过数据库查询session信息）总比做一次HMACSHA256计算 的Token验证和解析要费时得多.

【比较session和JWT】

相同点是，它们都是存储用户信息；然而，Session是在服务器端的，而JWT是在客户端的。

Session方式存储用户信息的最大问题在于要占用大量服务器内存，增加服务器的开销。

而JWT方式将用户状态分散到了客户端中，可以明显减轻服务端的内存压力。

Session的状态是存储在服务器端，客户端只有session id；而Token的状态是存储在客户端。

JWT的注意事项：

        消息体可以被base64解密为明文

        不适合存放大量信息

        无法作废未过期的jwt

页面加密:?

可以用eval加密下，不过解密也是一下下的事，总的来说前端无法加密，只能增加js代码阅读难度，如起一些没关联的变量名，a,b,c,d,e此类，如混淆

##### 在项目中什么模块使用？(如何实现票据刷新等场景)

我们在做<十次方>社交平台项目的用户登录模块使用到了jwt来验证用户信息.我们在登录成功之后生成token存储到cookie中,校验票据即可访问其他界面.

在我们做项目的时候碰到一个问题,我们的token是要设置过期时间的,如果一直是该用户登录,token过期后总需要让该用户重新验证身份信息,会导致用户体验度不高,所以我们在redis中设计了一个表里面存入了用户的id和ip,以及我们在生成token的时候自定义了一个属性来存储用户的ip,这样用户在登录的时候自动存入id以及ip,在token过期后可以判断该用户的ip是否和存入的ip相同,无需登录验证直接实现票据的刷新.

这样也解决了一个问题,因为我们的token是无状态的,我们的服务器是没有办法获取用户信息的,就是如果两个相同终端登录同一个账号,一个token还没过期,有生成另一个token,会导致用户信息的安全问题,在我们的token中存储了ip后,在生成token前可以去redis设计的表中去查询该id是否存在,如果不存在,则添加该用户的id和当前ip.如果存在,则修改该用户的ip为当前ip.生成的token中也存在有我们用户的ip,在第二个客户端登录了之后,第一个客户端的token就是错误的,就可以保证我们用户的信息安全.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

由于微服务大都是分布式的，需要几台服务器部署，当一个用户在其中一台服务器登录后，传统的方式是session保存其登录信息，然后可以使用共享存储共享，比如redis共享，这种方案的缺点在于共享存储需要一定保护机制，因此需要通过安全链接来访问，这时解决方案的实现就通常具有相当高的复杂性了，所以这里使用基于令牌的方式做登录。

JWT简介

简介网上都有，下面是摘抄的一部分，做做笔记。。。

JWT(JSON WEB TOKEN)是一个非常轻巧的规范，这个规范允许我们使用jwt在客户端和服务器之间传递安全可靠的信息。

JWT由3个部分组成，分别是头部、载荷、签名。

头部部分

{

“alg”: “HS256”,

“typ”: “JWT”

}

alg描述的是签名算法。

载荷部分

{

“iss”: “发行者”,

“sub”: 主题”,

“aud”: “观众”，

“exp”:”过期时间”,

“iat”:”签发时间”

以下可以添加自定义数据

“id”:”1”,

“nickname”:”昵称”

}

Base64算法是可逆的，不可以在载荷部分保存用户密码等敏感信息。如果业务需要，也

可以采用对称密钥加密。

签名部分

HMACSHA256(Base64(Header) + “.” + Base64(Payload), secret)

签名的目的是用来验证头部和载荷是否被非法篡改。

验签过程描述：读取Header部分并Base64解码，得到签名算法。根据以上方法算出签名，如果签名信息不一致，说明是非法的。

为什么使用JWT

前后端分离

以前的传统模式下，后台对应的客户端就是浏览器，就可以使用session+cookies的方式实

现登录，但是在前后分离的情况下，后端只负责通过暴露的RestApi提供数据，而页面的

渲染、路由都由前端完成。因为rest是无状态的，因此也就不会有session记录到服务器

端。

传统方式带来的安全性问题

在前后端分离通过Restful API进行数据交互时，如何验证用户的登录信息及权限。在原

来的项目中，使用的是最传统也是最简单的方式，前端登录，后端根据用户信息生成一个

token，并保存这个 token 和对应的用户id到数据库或Session中，接着把 token 传给

用户，存入浏览器 cookie，之后浏览器请求带上这个cookie，后端根据这个cookie值

来查询用户，验证是否过期。

但这样做问题就很多，如果我们的页面出现了 XSS 漏洞，由于 cookie 可以被

JavaScript 读取，XSS 漏洞会导致用户 token 泄露，而作为后端识别用户的标识，cookie

的泄露意味着用户信息不再安全。尽管我们通过转义输出内容，使用 CDN 等可以尽量避

免 XSS 注入，但谁也不能保证在大型的项目中不会出现这个问题。

在设置 cookie 的时候，其实你还可以设置 httpOnly 以及 secure 项。设置 httpOnly

后 cookie 将不能被 JS 读取，浏览器会自动的把它加在请求的 header 当中，设置

secure 的话，cookie 就只允许通过 HTTPS 传输。secure 选项可以过滤掉一些使用 HTTP

协议的 XSS 注入，但并不能完全阻止。

httpOnly 选项使得 JS 不能读取到 cookie，那么 XSS 注入的问题也基本不用担心了。

但设置 httpOnly 就带来了另一个问题，就是很容易的被 XSRF，即跨站请求伪造。当你

浏览器开着这个页面的时候，另一个页面可以很容易的跨站请求这个页面的内容。因为

cookie 默认被发了出去。

性能问题

如果将验证信息保存在数据库中，后端每次都需要根据token查出用户id，这就增加了

数据库的查询和存储开销。

Session方式存储用户id的最大弊病在于Session是存储在服务器端的，所以需要占用

大量服务器内存，对于较大型应用而言可能还要保存许多的状态，一般还需借助nosql

和缓存机制来实现session的存储，如果是分布式应用还需session共享。

兼容问题

在移动端app里，或者是前后端分离的架构中，用户访问的是前端的web server(如

node.js)，前端的渲染，ajax请求都是由web server完成的，这里就跟传统的不一样了，

用户不是直接访问后台应用服务器的，这时候用cookie+session就比较麻烦，问题在于

开发繁琐、安全性和客户体验差、有些前端技术不支持cookie(如微信小程序)

带来的好处

简洁，可以通过URL, POST 参数或者在 HTTP header 发送，因为数据量小，传输速度快。

自包含，负载中包含了所有用户所需要的信息，避免了多次查询数据库，服务端也不需要

存储 session 信息，做到了服务端无状态。

JWT方式将用户状态分散到了客户端中，可以明显减轻服务端的内存压力。除了用户id

之外，还可以存储其他的和用户相关的信息，例如该用户是否是管理员、用户所在的分组

等。

JWT能轻松的实现单点登录，因为用户的状态已经被传送到了客户端。

支持移动设备，支持跨程序调用，Cookie 是不允许垮域访问的，而 Token 则不存在这个

问题。

因为有签名，所以JWT可以防止被篡改

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1.讲讲jwt?

1. 搜索前台怎么才能搜出来东西？

搜索的其实是elasticsearch的索引，通过搜索其本身的索引

为啥使用angularJS？

1.一款非常优秀的前端高级 JS 框架

2.由 Misko Hevery 等人创建

3.2009 年被 Google 收购，用于其多款产品

4.有一个全职的开发团队继续开发和维护这个库

5.有了这一类框架就可以轻松构建 SPA 应用程序 single page web application

6.通过指令扩展了 HTML，通过表达式绑定数据到 HTML。

更少的代码，实现更加强劲的功能 Angular实现数字加一 ，Angular实现加法运算，Angular实现数据列表

二. AngularJS的意义带领前端进入MVC时代

MVC是Model - View - Controller的英文缩写。

模型(model)－视图(view)－控制器(controller)。

MVC是一种web应用开发模式。

三.MVC模式

Model（模型）是应用程序的核心，指的是程序在数据库中存储数据。

View（视图）是应用的界面，将数据库里的数据展现出来，展现给程序的使用者。

Controller（控制器）处理用户的交互行为，程序使用者更改数据，由控制器接受并发送给模型。

1. es支持跨域吗？

找到config文件夹下的elasticsearch.yml，在文件的末尾添加如下内容：

http.cors.enabled: true

http.cors.allow-origin: "\*"

1. 信息模块怎么做的？ 用的什么技术？

简单的信息添加和修改 所在省市用到了二级联动 就一个省市的二级联动；

1. 个人收藏这块怎么做的？

个人收藏其实比较简单，用户只有登录后才能进行收藏，是一个多对多关系；通过中间表把用户ID与相应内容id关联即可

1. 上传视频怎么做的？

上传图片其实就是把视频通过前台form表单，添加一个enctype="multipart/form-data"关键字，将文件转换为流的方式传送给后台；后台有相应工具类，将视频各种格式及存储位置都规定好；对视频大小有一定限制，上传失败或成功会返回一个相应参数提示；

1. 个人记录是记录的什么？

个人记录记录的是一个浏览足迹，用户看完整个视频或文章会做一个记录，存储最近三十条浏览记录展示给用户；

1. Feign有什么用？

Feign实现服务间的调用

Feign是简化Java HTTP客户端开发的工具（java-to-httpclient-binder），它的灵感 来自于Retrofit、JAXRS-2.0和WebSocket。Feign的初衷是降低统一绑定Denominator到 HTTP API的复杂度，不区分是否为restful。

@FeignClient注解用于指定从哪个服务中调用功能 ，注意 里面的名称与被调用的服务 名保持一致，并且不能包含下划线。

@RequestMapping注解用于对被调用的微服务进行地址映射。注意 @PathVariable注 解一定要指定参数名称，否则出错

1. Jkenns是用来干嘛的？

Jenkins是一个开源软件项目，是基于Java开发的一种持续集成工具，用于监控持续重复的工作，旨在提供一个开放易用的软件平台，使软件的持续集成变成可能。

Jenkins功能包括

1、持续的软件版本发布/测试项目。

2、监控外部调用执行的工作。

Jenkins的主要功能是将项目中重复执行的工作自动化的执行。如软件的建造（构建）和在配置文件下设置的job。具体功能大概有：

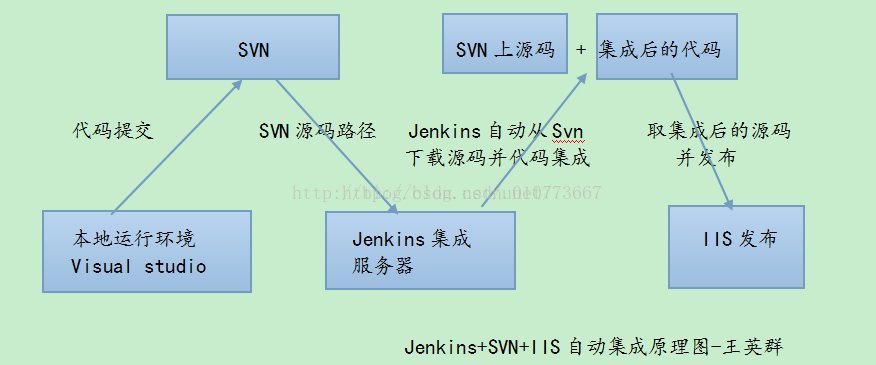
         1.软件的持续构建和测试，Jenkins提供了一个系统，使开发人员可以很容易的将改变集成到工程中。自动化的，持续的构建有利于提高开发效率。

         2.监视job的执行，如下图所示，job可以实现很多的功能，Jenkins可以对这些项目进行显示，让用户更清楚的注意到这些 损毁的job。（具体情况后文中介绍）

         3.项目源代码修改的检测，jenkins能够从项目的Subversion/CVS生成最近修改的集合列表，且不会增加Subversion/CVS Repository的负载。

         4.分布式构建，Jenkins可以将工程构建到多台机器，更好地利用硬件资源，节省时间。

         其他，Jenkins代码开源，用户可以自己编写插件。所以jenkins可以实现很多的功能。



##### 短信微服务这块

为什么使用MQ做短信服务?

解耦 共用性 并发量

代码实现讲讲?

因为我们项目是一个微服务的系统，有好多工程都需要使用发送短信的功能，比如说 注册、支付还有很多场景都用到了发短信的功能，我们采用的架构是SOA 面向服务的架构，而短信不涉及具体业务，而是一个通用的功能，所以我们项目中,是采用SpringBoot 和 ActiveMQ实现的一个微服务抽取出所有发送短信的功能。短信我们使用的是阿里大于提供的短信服务。阿里的大于短信微服务接口,是现在一块很不错的短信服务这一块

##### 举例说明:

我们在用户注册这块就用到了短息验证,在填写完其他的信息完时,这时候会有一个手机号验证,来验证我们的用户手机号是否可用,和本人是否一致,这样呢我们就会用到一个短信的验证技术,我们用的是一个阿里的大于短息接口,但是用到阿里的短息接口的话,我们用的是ssm的一个底层框架,但是阿里短信 接口是一个微服务,所以我们用到了springBoot这个微服务框架来搭建短信接口,这大体就是我们这个短信微服务的一个简绍

##### 在业务这一块呢

我们先是在注册的一个页面,进行一个短信验证码的发送,调用后台短信这块的一个方法这个方法在被带调用后,在后台生成短信验证码,然后通过消息中间件发送我们搭建的sms短信微服务的这个服务器上,进行一个短信的发送,这个就是我们短信的发送一个业务流程,

为了实现这一块的业务,

我们用到了我们搭建的springBoot微服务 sms的短信接口,还有JMS中间件activeMQ的消息队列,服务端的话会用到用户中心这一端,短息微服务的这一端,并在这俩端搭建AMQ消息队列进行通信,表呢注册时用到了一个用户表,

##### 技术的介绍

阿里的大于短信微服务借口,是现在一块很不错的短信服务这一块,对于我们开发这一块来说阿里的大于是很不错的选择,因为对我们这些,中小型的项目来说,不仅完美的支持,还融合了三大运营商的通信能力.符合我们现在项目的一个支持,springBoot呢,是一个轻量级的服务框架,因为我们的阿里的大于也是一个轻量级的服务,相对于spring来说,太多的配置文件,和重量级的配置,比较繁琐,和承重.并不适合我们来做短信服务这个端口,所以呢.我们就用了springBoot.

activeAMQ这一块呢,是消息中间件,是用来管理消息队列的一个服务,activeAMQ,有俩种模式,一种一对一,点对点的模式,另一种呢是一对多的一种模式,一个人发送,有多个人消费

##### 具体的细节

当我们的后台发送短信的方法被调用的时候,

首先,我们会判断一下手机号码是否规范,这里,

我们会写一个判断手机是否正确的正则工具类,

然后调用工具类来判断手机号是否正确,

这里,如果手机号不正确的话,直接返回到前台,并把错误信息告诉用户

如果手机号正确的话,哪们我们调用发送验证码的方法,

##### 发送验证码

<1>在发送验证码的时候,

首先,我们会生成一个6位数的验证码,

在生成验证码的时候我们没有用到直接用随机数\*6位来生成验证码,

因为这样我们的验证码会有缺失.这里我们自己写了一个生成验证码的随机数,

首先呢我会定义一个0-9字符串,然后呢在定义一个stringBuffer对象,

和随机数的对象这样我们用循环stringBuffer这个对象进行操作追加,

循环次数呢是6，追加的对象是用随机对象生成一个0~10的下标,

用下标来获取定义0~9那个字符串中的一个字符，

然后把这个追加后的对象toString后返回,这样我们验证码就写好啦，

在生成验证码后,我们还要把验证码的存入到redis里,

为添加时验证验证码是否正确

<2>redis这块,

我们用的是RedisTemplate<String , Object> redisTemplate这个对象来操作redis,

redis这里呢,我们用的是hset的这个方式存的值,给了一个key, find 值,并设置了失效时间

<3>然后就是利用JSM来把们的验证码发送sms服务器，

这里我们的jmsTemplate这个对象来操作AMQ发生短信,

在这里呢,我们把要发送的信息放到MapMessage里,这里的参数,

有待发送人的电话号码,有阿里大于的模板钥匙,和我们生成的验证码,还有一个是我们要用的模板类型.

这样把 MapMessage 这对象返回，这样呢,

完成了我们用jsm技术来发送到sms的一个操作,

##### 收验证码

接下来呢是我们的sms用jsm的AMQ来接收我们刚刚发出的消息.

在sms里呢,我们通过 @JmsListener 这个注解来接受发生的消息，

在收到消息后我们把消息放到我们的短信服务工具类里,

这个短息工具类呢,我们需要配置一些东西的accessKeyId accessKeySecret我们放入的数据，

有待发送人的电话,使用的短信签名,短信模板,还有就是模板变量的值。

设置好这些后,进行发送，并在控制台打印发送状态.这样我们就完成了一个短信发送的操作。

最后当我们用户收到验证码后,并把验证码填写到验证码的框里,进行注册

，在后台当我们收到前台发来的注册信息后,

我们要从redis里取出我们刚存的验证码和用户刚刚前台填写验证码进行对比和判断,

一样则注册成功,不一样注册失败.

这样,我们就完成了一个短信验证注册.

商品搜索

搜索采用elasticsearch 好处?

**Elasticsearch与Solr的比较**

1、当单纯的对已有数据进行搜索时，Solr更快。

2、当实时建立索引时，Solr会产生io阻塞，查询性能较差，Elasticsearch具有明显的优势。

3、随着数据量的增加，Solr的搜索效率会变的更低，而Elasticsearch却没有明显的变化。

Elasticsearch与Solr的比较总结

Solr利用Zookeeper进行分布式管理，而Elasticsearch自身带有分布式协调管理功能

Solr支持更多格式的数据，而Elasticsearch仅支持json文件格式

Solr官方提供的功能更多，而Elasticsearch本身更注重于核心功能，高级功能多有第三方插件提供

Solr在传统的搜索应用中变现好于Elasticsearch，但在处理实时搜索应用时效率明显低于Elasticsearch

Solr是传统搜索应用的有力解决方案，但Elasticsearch更实用于新兴的实时搜索应用。

具体操作配置?

ik分词器应用?

保证es与数据库的同步方案：

1、首先添加商品入数据库，添加商品成功后，商品入ES，若入ES失败，将失败的商品ID放入redis的缓存队列，且失败的商品ID入log文件（若出现redis挂掉，可从日志中取异常商品ID然后再入ES），

task任务每秒刷新一下redis缓存队列，若是从缓存队列中取到商品ID，则根据商品ID从数据库中获取商品数据然后入ES。

消息队列实现增量同步的方式，是在主服务对数据库进行创建、删除、修改一条记录时，发布一条主题消息给消息队列，同时同步服务需要订阅相关主题，这样消息队列就可以将更新的记录转发给同步服务，同步服务再根据消息的内容在ES里面进行更新记录。

========================================================================

消息队列实现增量同步除了可以解耦主服务和同步服务，还有一个好处就是保证同步的容错性，比如当数据库添加一条记录时，如果直接采用HTTP的方式（可能是一个post请求）与同步服务取得联系时出现连接失败、post请求失败的时候，如果不采取任何措施这条记录就会无法得到同步。而消息队列的失败重发的机制可以很好的解决这个问题，同时消息队列，FIFO（先进先出）的机制也保证了消息转发的顺序。

我们平时如果搜索一些东西的时候，一些小的系统，简单的数据库就用like来进行查询

关键字就够了，但是医学慕课平台这种数据量还是比较多的，所以说要想勾客户

的胃口，吸引他们的注意力的话，我们在他检索商品，检索他想要的东西时肯定要给他展

示更快、更精准的方法，所以我们就用到了Solr引擎。他其实就是基于Lucene开发的一

种全文搜索引擎。不过好像也有公司在用同级别的ElasticSearch,简称es。Es也是一款

基于lucence的搜索引擎。

Solr跟Es相比：

solr查询快，更新索引相对慢

es更新索引快，查询相对慢。

**在做这个搜索模块的时候 我们用到的是ES搜索引擎 ：（Elasticsearch是一个实时的分布式搜索和分析引擎。是当前流行的企业级搜索引擎。设计用 于云计算中，能够达到实时搜索，稳定，可靠，快速，安装使用方便**

**）**

**我们也想到了用solr 但是通过对比 发现ES更符合我们这个项目的需求**

**因为ES它在实时搜索的时候效率是高于solr的 Solr更适合传统搜索应用在搜索大的数据的情况下 ES的效率会很高 solr需要配域而ES不需要 相对来说比较简单**

**我们在做搜索的时候主要是根据用户的需求进行课程搜索，我们是使用docker搭建一个ES的公共服务**

**一开始我们是用postman调用restapi新建索引和文档 在新建文档的时候 id是由系统自动生成的。在修改文档的时候是根据id来进行修改的 如果id不存在 它就会创建新的文档 删除文档的话也是根据id来进行删除的 通过这个我们发现 如果都是通过rest请求的方式使用ES未免太过麻烦，而且也不够人性化。然后就想到了使用图形化界面来实现ES的日常管理，最常用的就是Head插件 在安装好Head插件后 我们发现了一个问题 就是它在启动之后 点击连接按钮没有反应 按F12发现了一个错误 “No 'Access-Control-Allow-Origin' header is present on the requested resource” 这个错是由于ES默认不允许跨域调用，而es-head是属于前端工程，所以报错。**

**然后我就修改了ES的配置 让它允许跨域调用**

**我们在做测试的时候 发现它的分词不太智能 默认的中文分词是将每个字看成一个词 这样是不符合要求的 所以我们就用了IK分词器来解决这个问题 IK 分词器有两种算法 其中ik\_smart为最少切分，ik\_max为最细粒度划分**

**我们还可以自定义词库**

**然后 我们又要考虑一个问题 因为我们查询的数据是来自不同的数据库或者表 这样的话就很难以一个比较快的速度去查询数据 这个时候我们就想到了用这个logstash进行同步数据 同步数据库与索引库 （solr是用的定时器 而ES用的是logstash）logstash是一款轻量级的日志搜集处理框架， 咱们在公司里边用这个ES也是分两种 一种是增量索引更新一种是全量索引更新 当我们安装好logstash 需要在logstash的bin文件夹下创建一个.config结尾的文件，在里面配置mysql的链接还有查询课程表的sql语句、es的链接，最主要还是需要配置一个定时任务。这个定时任务就是精确到年月日时分。在这儿我配的是每1小时同步一次。 可以让它定时的去读我的数据库里边的东西，读完之后再把他所读东西搜集起来指定给ES 这样的话就相当于隔1小时ES就被同步一次**

会员充值（支付宝支付）：

支付宝接口具体流程?

遇到哪些问题?

需求: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.

订单支付的方式的有很多种,用的比较广泛的有微信和支付宝,我们前期主要用的是支付宝支付.

那么使用支付宝支付呢,我们在开发阶段主要使用的支付宝的沙箱环境,这是支付宝给我们开发人员对外的一种模拟测试的环境,功能方面和支付宝大致没区别.

在支付宝沙箱后台会有一个应用,这里边有几个配置信息,需要我们注意,因为在项目中我们会用到这几个配置的信息,一个APPID,一个是支付宝网关,这个网关地址中会含有dev三个字母,这说明我们是在沙箱环境中,还有RSA2的加密方式,这里的公钥可以通过支付宝提供的RSA签名工具生成一对公钥秘钥,这里生成的是商户的公钥和秘钥,将商户的公钥放到沙箱后台商用应用公钥中,支付宝会自动匹配出一个支付宝公钥与之对应,另外我们还需要配置一些信息,如支付网关,return-url和notify-url.

return-url:指的是支付宝完成支付之后要跳转的一个页面,所以我们提供的这个url主要用于显示支付结果

notify-url:指的是支付完成后支付宝还要做一个异步的通知,这个接口要求我们处理结束会必须有结果返回,不然支付宝会在25小时内调用8次.

我们在这个路径中会进行订单状态的修改,因为这个方法有可能会出现多次,所以我们也要进行去重的处理.具体使用时,用户点击结算订单,我们获取到订单号,支付金额,订单名称,描述信息,提交到我们的后台方法中,在后台直接使用支付宝的SDK就可以完成支付.

具体实现:

因为我们是分布式项目.所以给支付宝支付创建一个服务端,在服务中会有一个方法用于生成alipay的form表单提交,在生成的这个form表单中,有一个script标签会自动提交生成的这个from表单,我们进行了一个小处理,就是将这个script标签去除掉,通过用户在页面上点击支付按钮,在触发form表单提交,这里用到了String类的replace方法,同时我们还需要将订单号和金额返回给前台,订单号我们是使用一种雪花算法生成的,雪花算法是用于分布式高并发下唯一标识的算法,它是由,时间戳,工作机器ID和序列号和一个不用的第一位组成的64位的序列号,它的最大并发量可以在同一秒中自增生成26万个不冲突的可排序的id.在之后这里还使用了一个map集合,将需要返回的数据,全放到这个map集合中,将装好数据的map集合返回给前台.当用户点击订单提交的时候,跳转到支付页面,支付页面会进行初始化,调用后台生成form表单的这个方法,将map中的数据显示在页面中,这里会有一个小问题,就是我们生成的表单是一个字符串,这是由于我们前台使用angularjs,出于安全机制,它会认为你这个数据是一个被注入的不安全数据,就会直接进行展示,所以我们需要用到angularjs的一个sce服务,在js中加入一个filter服务,将这个form表单进行一个过滤处理,这时候就可以正常生成from表单,点击支付按钮以后就会跳到支付宝的支付页面进行一个扫码,就可以完成支付了.

接下来是完成支付后的操作,首先需要配置一下return-url和notify-url,return-url中我们直接配置支付成功后跳转的支付成功页面，而notify-url我们需要配置成触发后台的一个方法,这个方法主要用于修改订单的状态.在这个方法中我们可以通过request获取支付宝给我们返回的参数,里边的参数我们可以通过支付宝文档中可以去查看都返回什么参数,支付宝还提供了验签的方法,我们只需要将支付宝提供的这个方法放到我们的服务当中就好,这个方法中有四个参数一个是前台传过来的map数据,一个是支付宝的公钥,一个是支付宝中的charset,还有一个就是签名的类型,这个方法会报一个异常我们只要trycatch就好了,成功就返回true,失败就返回false,在控制层我们会判断服务层给我们返回的Boolean值,如果为true就说明验证签名成功,我们就获取这个订单号,判断系统中是否有这个订单,在判断一下支付的金额是否一致,在获取交易状态,如果这个交易状态为trade\_success,就说明支付成功了,我们就将订单的状态改为已支付状态,并返回一个succes字符串,因为notify-url只认识success这个字符串,说明交易成功,不然都认为你交易失败.

上传课程模块：

上传用的技术?

上传到哪?

具体实现?

遇到问题?

**图片防盗链原理及实现?**

我的实现防盗链的做法。基本原理就是就是一句话：通过判断request请求头的refer是否来源于本站。（当然请求头是来自于客户端的，是可伪造的，暂不在本文讨论范围内）。

首先我们去了解下什么是HTTP Referer。简言之，HTTP Referer是header的一部分，当浏览器向web服务器发送请求的时候，一般会带上Referer，告诉服务器我是从哪个页面链接过来的，服务器籍此可以获得一些信息用于处理。比如从我主页上链接到一个朋友那里，他的服务器就能够从HTTP Referer中统计出每天有多少用户点击我主页上的链接访问他的网站。（注：该文所有用的站点均假设以 http://blog.csdn.net为例）

假如我们要访问资源：http://blog.csdn.net/Beacher\_Ma 有两种情况：

1． 我们直接在浏览器上输入该网址。那么该请求的HTTP Referer 就为null

2． 如果我们在其他其他页面中，通过点击，如 http://www.csdn.net 上有一个 http://blog.csdn.net/Beacher\_Ma 这样的链接，那么该请求的HTTP Referer 就为http://www.csdn.net

知道上述原理后，我们可以用Filter去实现这个防盗链功能。网上的做法多是用列举的方式去做的，而我这里是用正则去做，相对比较灵活点，另外，我效仿了Spring的filter做法，加了个shouldBeFilter的方法，考虑到，比如假如你要拦截\*.Action的一部分方法，而不是全部时，我们就可以先看看请求的URL是否shouldBeFilter，如果不是的话，那么就直接放行，在效率上有所提高。

上传课程模块就是用户对一些医疗课程视频的上传功能，包括标题、封面、介绍、科室、头像、视频等，其中对于上传的封面图及头像有大小（2MB）限制，视频限制为100MB内；前端使用 angularJS 异步上传，后端使用 springmvc 的 MultipartFile 类型来接收，放到分布式图片服务器中，服务器返回图片路径把路径返回页面回显图片。用户上传成功并不能马上公布在平台上，需要经过后台管理员审核通过后，在同步到搜索上。

个人信息模块：

个人收藏怎么做的?

个人记录怎么做的?

分为个人资料、个人收藏、个人记录。个人资料是个人信息的一些修改与添加，主要是信息的完善操作；里面有收藏模块，用户收藏的一些内容可以保存于收藏夹中；上传课程记录及学习记录等的一些记录展示；个人资料主要就是个人信息的修改，省市二级联动；个人收藏是对喜爱的视频的一些收藏，就是对相应内容根据id加一个状态标签；个人纪录是对相关视频学习浏览的一个记录，可是说就是一个浏览痕迹，对看完的视频或内容做一个标记，取得最近三十条或二十条记录进行展示；