**东方校园网**

负责模块及实现：

1. 权限模块：

注册

短信验证码

在我们的项目当中很多地方都用到了我们的短信验证的功能

举例说明: 我们在用户注册这块就用到了短信验证,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。

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中间件的RabbitMQ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RabbitMQ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，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.

RabbitMQ这一块呢,是消息中间件,是用来管理消息队列的一个服务,RabbitMQ,有五种模式,

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,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用到的是直接用随机数\*6位来生成验证码,为了避免，验证码会有缺失.这里我们自己定义了一个最大数和最小数，随机生成完毕后，对验证码进行判断，如果验证码小于最小数，n那么我们就让生成的验证码加上定义的最小数来防止验证么丢失。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hash的这个方式存的值,给了一个key, value值,并设置了失效时间

<3>然后就是利用RabbitMQ来把们的验证码发送sms服务器，这里我们的rabbitMQTemplate这个对象来操作RabbitMQ发送短信,在这里呢,我们把要发送的信息放到Map里,这里的参数,有待发送人的电话号码,还有阿里大于的模板钥匙,还有我们生成的验证码,还有一个是我们要用的模板类型.这样把 Map 这对象返回，这样呢,完成了我们用RabbitMQ技术来发送到sms的一个操作，接下来呢是我们的sms用RabbitMQ来接收我们刚刚发出的 消息.在sms里呢 ，我们通过 @RabbitListener 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和 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.

## **token应用流程？**

初次登录：用户初次登录，输入用户名密码

密码验证：服务器从数据库取出用户名和密码进行验证

生成JWT：服务器端验证通过，根据从数据库返回的信息，以及预设规则，生成JWT

返还JWT：服务器的HTTP RESPONSE中将JWT返还

带JWT的请求：以后客户端发起请求，HTTP REQUEST HEADER中的Authorizatio字段都 要有值

**登录模块**   
JWT简介  
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。简单理解可以就是一个 Token 值，只是这个 Token 值比较特殊，有两个 . 号分隔。

1. 兼职模块：

我们的兼职模块，主要是面向学校内的学生，为学校的学生提供一个锻炼自己的平台。

进入界面我们可以看到一个分类，校内还是校外，以及全部，在下面是一个大的标题，标题下面是一个小的简介，还有一个过期时间，点击进去，会看到兼职信息，以及兼职联系的联系方式，以及兼职工作的描述。

因为我们这个平台是面向校内学生的，所以我们要确保好学生的安全，以及其他问题，所以问在发布兼职的时候，会有一个担保人，担保可最低是校内在职老师，如果老师离职的话，平台会将其老师担保的所有兼职信息全部关闭，再发布兼职的时候，会选择区域（校内，校外），兼职类型，兼职描述，以及发布人的联系方式以及其相关信息，并且需要把担保人的账号填写到担保人的位置，这样他在提交的时候，后台会通知担保人，并让其确认，如果担保人状态异常，比方说离职，或者请假，就会驳回，提示发布人此担保人暂时无法担保，如果担保人状态正常，会让其授权获取自己的相关信息，并调用短信接口给其担保人注册平台时的手机号进行短信验证码发送，这样是为了验证其手机号是否属于正常通信状态，并且是否是担保人本人，当担保人担保之后，兼职信息将会出现在学校的兼职平台上。再发布兼职的时候，需要输入有效期，一是因为有的兼职他会在招聘一段时间后，不缺人了，就会停止招聘，但是平台上还有相关信息，学生进行报名之后，就无反应，二是为了不占用我们数据库的内存，所以我们选择了进行删除处理，我们这个兼职因为数据量大，我们这里使用的是redis缓存技术，并设置过期时间，当有效期时间到了我们会进行直接删除，清空记录。如果兼职信息与实际信息不符，我们还提供了匿名举报，当管理员收到了 举报理由后进行审核，当审核通过，我们将会联系相关担保人进行处理。

1. 搜索模块：

利用ElasticSearch搜索引擎实现搜索模块。通过安装Elasticsearch安装，能够调用RestAPI完成基本增删改查操作，Head插件的基本使用、IK分词器进行分词以及 完成mysql与Elasticsearch的同步工作。

**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。

在项目的搜索模块我们用到了es，主要是对用户发表的文章进行海量检索，个人感觉es的使用还是比较简单的，我们使用docker搭建了一个el的公共服务，然后我们使用spring-data整合es完成文章的发布与浏览操作。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，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。（logstash同步数据.....)

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。(详细见下文)

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24小时同步一次。它有点儿类似于一个定时器 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24小时ES就被同步一次

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。

1. 讨论模块：

首先呢，我们这个模块有这么一个发布资讯的功能，我们在做这个发布资讯的时候，首先考虑到，里面不能有一些不良信息，比方说，反抗党，玷污国家等等的词汇，语言，所以我们有这么一个机制，就是你在发布之后，有审核这么一个过程，一般在两小时之内审核完毕。我们在发布模块的时候，首先想到了这个隐私性，我发布人的名字可以和登陆名字不一样，就好比我们qq看点一样，里面的名字，和你的昵称不一样，所以我们这个发布人的名字，可以和登录名不同，我们的文件支持单张图片上传，在这里我们调用的是一个工具类，实现了上传，并且给其实现了点击下载功能。发布提交之后，我们这条文章会传入到redis里，进行一个缓存，这是我们考虑到这个数据库压力这个问题，在审核通过之前，我们这个文章，资讯就暂时存储在redis里，如果审核不通过，将会通知发布人，一天之后进行删除，请及时修改重新提交，并且把不通过的原因一起反馈给发布人，如果通过，就会把此文章添加到数据库里。在发布成功之后，我们的文章会出现在校园网的资讯模块里，上传的那张图片，会出现在文章的头部，当我们点击进去之后，可以看到我们我们的发布人名称，文章类型，在文章的尾部，有这么一个评论框。

评论框可以进行实时评论，我们这里使用的是SpringDataJpa进行发布评论，因为是实时的，所以当你评论完，会立即出现在文章的下方，但是有时评论里会出现不良言论，此时除了我们管理员可以删除以外，发布人也可以删除文章的评论，其他人可以举报评论，提交举报理由，由后台管理员进行审核，并处理。在评论了文章之后，用户的评论可以被其他用户评论，当用户在评论文章的时候，数据库里保存的是文章id，但是其他用户评论用户了之后，那个被评论用户的id就会变成文章id，也就是把此用户发布的评论变成一个小文章，并对其进行评论，并且可以进行回复，当此条评论删除时，就好比文章被删，下面相应的评论也会被删除。大致结构就是这样，但是我们这有个问题，就是我们这个评论有几个特点，数据量大，写入操作频繁，价值较低，而且如果任其写入数据库的话，数据库压力会很大，所以我们在这里使用到了mongodb用来存储相关数据使用mongoDB更能保证用户的访问速度，文档结构的存储方式，能够更便捷的获取数据，高效存储二进制大对象 (比如照片，）支持丰富的数据表达,支持的查询语言非常丰富，适合大数据量的存储，依赖操作系统VM做内存管理，吃内存也比较厉害，同时MongoDB主要解决海量数据的访问效率问题

首先mongodb是一个开源的nosql数据库，它与传统的关系型相比，在数据存储方面扩展性很好，并且在数据类型上他没有采用传统的关系型数据库那样，将表结构设计死，在创建了collection后是不需要定义数据结构，它的数据结构采用更加灵活的Bson格式，bson就是在json的基础上做了扩展，比如加入了对日期Date和BinData类型等类型的支持。

并且mongodb也提供了索引查询支持，检索速度也很快，并且mongodb支持副本集集群、分片集群的搭建，在我们项目中就是采用副本集的集群方式，我们采用了3台机器搭建mongo集群，因为mongo副本集支持故障转移，当一个主节点挂掉后，又会在剩下的两个从节点中选举出一个“主节点”继续工作，保证了系统高可用，而且在后期保存数据的过程中，mongo支持动态的添加或者删除节点，非常方便维护。Mongo最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立[索引](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95" \t "https://baike.baidu.com/item/mongodb/_blank)。(高性能、易部署、易使用，存储数据非常方便，支持完全索引，支持动态[查询](https://baike.baidu.com/item/%E6%9F%A5%E8%AF%A2" \t "https://baike.baidu.com/item/mongodb/_blank))，另外我们的文章有的评论会很多，比如几千条评论，我们不能一下子把数据库全部查出来吧，所以我们这里用了物理分页，将其根据ID排序后，先显示十条，看到第十条的时候，自动显示查询后10条，以此类推

1. 学员档案模块：

### 学习概况：设置一个基于注解（@Scheduled）定时器，每到一段时间开始运行代码，重新查询，根据时间区间统计在某一个时间段的总体情况，能够看到学员的项目所获得总学分和总学时，以及学习时长信息，我的成绩，学生查看自己的各项成绩

### 我的证书：学生点击我的证书，可以看到学生自己的各项证书情况，点击“证书预览”就会向后台通知，返回前台，就会看到每一项证书的预览情况，学习记录，分两类显示项目学习记录和公开课程学习记录两类