叫啥 年龄 上家公司 经历过什么项目 说一遍个人优势 工作中怎么怎么样 生活中怎么怎么样

自我介绍

您好，我叫高继夫，今年24岁，属猪的，我上家公司是在宜天信达（北京市朝阳区百子湾路16号百子园4号楼11层），目前做过彩票网站、电商、教育、oa系统这方面的，最近刚做完的是一个教育平台，主要针对的是中，小学生的一个学习网站，项目整体的框架使用的是最近新兴的微服务架构。**开发的所有环境都是基于docker镜像。Springcloud使用Eureka服务发现组件实现微服务的集中化管理，用feign实现服务间的调用。通过Hystrix熔断器来避免服务的级联故障。使用Elasticsearch实时分布式搜索分析引擎。最后运用Jenkins实现持续集成，使用zuul做对接前台的api网关，对缓存redis、mongodb进行处理。注册短信调用阿里大于接口，搜索课程docker完成项目的无缝对接。**项目的整体模块有首页、课程选择模块、常见问题模块、个人中心模块等等。我负责了课程选择模块，一般的课程是免费的，但是其中有一个一对一辅导的功能是付费了，在这里我调用了支付宝接口。阿里配的一个工具类提供了APPID公钥私钥的回调地址，参数有编码，金额等等信息传完后支付宝会自动调取一个支付页面。

# 时间推算

工作年份：2015-9

出生日期：1995

大学：2011-9~2015-7 17岁~21岁

高中：2008-9~2011-7 14岁~17岁

初中：2005-9~2008-7 11岁~14岁

小学：2000-9~2005-7 5岁~11岁

# 离职原因

：家里出了点事，因为回家的时间比较长，不想耽误项目的进度，所以就离职了。

对上家公司的评价：其实很感谢上家公司，因为从毕业就开始在公司工作，这几年能力的提升是离不开上家公司对我的帮助和提携的。不管是从技术方面还是做人方面，都让我成长不少。而且和同事以及经理之间的关系都比较融洽，这次离职，经理也是多次的劝说和挽留。但是这次家中有事，经理也不好多留，所以就辞职了。

主动提问：咱们公司最近在做什么方面的项目？

# 优课堂———————————————

## 课程支付模块

### 收款

支付宝支付的流程：要使用支付宝支付，必须要有支付宝的企业号，在企业号的后台可以创建一个应用（企业号的申请需要公司的相关材料，如组织号，营业执照等等），有了应用后可以查看应用的APPID，我们还需要使用支付宝提供的RSA签名工具生成一对密钥（商户私钥和商户公钥），商户私钥是要用在项目中，支付宝支付的配置上面，商户公钥上传到支付宝后台的应用上后，会得到支付宝公钥，这个公钥也要用到项目中，另外在项目中还需要配置，如签名方式，支付网关，及return-url和notify-url，return-url是支付宝在完成支付后要跳转的页面，所以我们提供的这个url主要用于显示支付结果，notify-url是支付完成后支付宝要做的异步通知，这个接口要求我们处理结束后必须有返回结果，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，首先要创建一个客户端对象AlipayClient（传入配置中的参数，appid私钥、公钥、签名类型），客户端创建后需要封装请求，Alipayrequest在请求中，传入return-url和notify-url及前台传入的参数，最后使用alipayclient的pageExexcute。getBody方法进行页面的提交，支付会跳转回我们注定的return-url路径，

我们在这个页面展示支付结果。同时支付宝还会调用我们的notify-url方法，并带上流水号，订单号，价格返回（其他参数还有好多，不一一列举，可在API中查看），此时你action的回调方法里回把这些数据进行比对并验证签名（支付宝提供的SDK可进行签名验证），都正确后给支付宝发送一个success，如果支付宝接不到返回会在25小时内调用8次。

遇到的问题：商户没有收到支付结果的通知问题？用户在我们的支付成功，支付成功一定收到支付宝的通知。在这个通知之前，是会回调我们提供的接口。支付宝回调的过程中，没有调通我们的支付接口。【支付平台不健壮，就是服务不稳定导致的】用户可以马上投诉，解决办法，客服后台，根据用户提供的订单，查询支付的信息（这个查询我们是直接调用支付宝接口的查询功能进行查询的，如果存在的话就是我们网站的问题，这我们的代码回修改订单的状态），当然我们还有一个对账系统，这个对账体系只要做的是，将我们支付平台的订单到第三方进行支付查询，看是否有支付计录。安全问题：签名机制 参数信息（密钥）---根据加密的算法进行加密---生成加密的签名---进行传递----我会根据密钥+传过来的参数，在进行一次加密也会生成一个签名，传过来的签名和我自己生成的签名进行对比，如果一直说明数据没有修改，如果不一致，说明数据修改了。

### 退款

1. 引入支付宝SDK

在这里说一下，其实每种支付都会有一个官方带的SDK，我们可以把这个sdk引入到我们的项目中去，然后我们就可以调用里面的方法和对象，给里面传值，进行我们相应的请求的访问和介入。

IMG_256

首先引入如上图所示的jar包，然后我们就算是成功了一半了。下载地址：

[https://docs.open.alipay.com/54/103419/](https://docs.open.alipay.com/54/103419/" \t "https://blog.csdn.net/tr1912/article/details/_blank)

         然后我们需要把这个jar包引入到所需要的地方，这里不做过多的描述。

二、找到需要的资源

        做支付，最最基本的还是各种appid，还有公钥，私钥，以及请求的网关等等内容，支付宝需要的包括：

三、编写代码

调用AlipayTradeRefundModel这个方法

APP\_ID：开放平台账号申请的appid

requestUrl：实现功能所需要请求的网关地址

APP\_PRIVATE\_KEY：用户自己生成的私钥

ALIPAY\_PUBLIC\_KEY：支付宝生成的公钥

CHARSET：我们需要的字符编码，一般为utf-8的

1. 编写请求代码

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import com.alipay.api.AlipayClient;

import com.alipay.api.DefaultAlipayClient;

import com.alipay.api.domain.AlipayTradeRefundModel;

import com.alipay.api.request.AlipayTradeRefundRequest;

import com.alipay.api.response.AlipayTradeRefundResponse;

public class AlipayUtil {

    private static final Logger LOGGER = LoggerFactory.getLogger(AlipayUtil.class);

    private static String APP\_ID = "234789720984098";

//支付宝退款请求的网关

private static String requestUrl = "https://openapi.alipay.com/gateway.do";

//用户自己生成的私钥

private static String APP\_PRIVATE\_KEY = "随机字符，见文档";

//支付宝公钥

private static String ALIPAY\_PUBLIC\_KEY = "从支付宝中获得";

//编码级别

  private static String CHARSET = "UTF-8";

  public static String refundOrder(AliRefundTO aliRefundTO){

System.out.println("开始调用支付宝加密\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//实例化客户端

AlipayClient alipayClient = new DefaultAlipayClient(requestUrl, APP\_ID, APP\_PRIVATE\_KEY, "json", CHARSET, ALIPAY\_PUBLIC\_KEY, "RSA2");

//SDK已经封装掉了公共参数，这里只需要传入业务参数。以下方法为sdk的model入参方式(model和biz\_content同时存在的情况下取biz\_content)。

AlipayTradeRefundModel refundModel = new AlipayTradeRefundModel();

refundModel.setTradeNo(aliRefundTO.getTrade\_no());

refundModel.setRefundAmount(aliRefundTO.getRefund\_amount());

refundModel.setRefundReason("商品退款");

//实例化具体API对应的request类,类名称和接口名称对应,当前调用接口名称：alipay.trade.app.pay

AlipayTradeRefundRequest request = new AlipayTradeRefundRequest();

request.setBizModel(refundModel);

try{

AlipayTradeRefundResponse response = alipayClient.execute(request);

System.out.println(response.getMsg()+"\n");

System.out.println(response.getBody());

}catch(Exception e){

e.printStackTrace();

LOGGER.error("支付宝退款错误！",e.getMessage());

}

return "";

}

public static void main(String[] arg){

AliRefundTO aliRefundTO=new AliRefundTO();

aliRefundTO.setRefund\_amount("0.01");

aliRefundTO.setTrade\_no("2018050421001004310518105890");

String res=refundOrder(aliRefundTO);

}

}

我在支付页面隐藏了一个表单，它是用来装支付宝支付的一些参数，我是封装一个from 表单，当我点击支付的时候，吧from表单中的参数传给后台。

在后台我创建了一个aliclient订单支付的客户端，这个客户端初始化的时候需要参数(这个参数在配置文件中定义好了，例如：要支付的网关路径，应用ID，私钥 和公钥）

初始化完成后需要封装一个请求对象，我是把同步的url、异步的url，付款金额，订单名称全部封装到请求对象中去，用alipayclient的客户端执行请求方法（alipay.pageEcute().getbody()）进行一次请求调用，请求调用后会返回一个界面，这个界面跳转到支付宝支付界面，这个时候就开始交钱了。

同步：

调用后台方法获取支付宝api中trade\_status（交易状态）、trade\_no（支付宝交易号），支付宝支付 成没成功 直接关系到我们这一次状态（1.trade\_closed[未付款交易超时关闭] 2.trade\_success[交易支付成功]）。在页面上我会判断一下他是否是trade\_succerss

trade\_no（支付宝交易号）是为了我们以后的退款、查看商品的状态

支付宝在处理完之后会吧页面跳转到return\_url

异步的调用我后台的方法就是notify\_url就是把订单状态修改了

根据支付宝api异步通知

将异步通知获取的所有参数都存到map中去，进行验证。验证有两种结果：

验证成功，说明是一个合法的请求，就根据里面的参数进行处理。把订单的状态进行修改 还需要把支付宝交易号存入到系统中，同时做一个去重的处理，例如：支付宝调用多次succerss，处理完之后返回trade\_success。

首先微信提供了一个官方的微信公众号平台，我们申请成功以后，可以在这个平台上直接进行配置。比如消息回复，有消息自动回复，被关注时被动回复、关键字回复等。回复的消息包括文字消息、图片消息、语音消息、视频消息等。还有就是自定义菜单，最多可以配置3\*5个菜单，点击菜单可以回复消息，跳转链接，跳转小程序。如果想要进一步使用更复杂的功能，就需要成为开发者，调用他的api接口文档了。微信公众号的接口都是restful风格的接口，我们可以用httpclient直接调用，接口有些是json数据格式，有些事xml数据格式。调用接口文档首先需要接入测试和网页授权。接入时，必须使用域名和80端口，在公众号平台配置好url，发送请求，平台url里边接收参数，按照要求生成签名，与传过来的签名进行比较，验证签名，如果成功返回一个字符串。这样就接入成功了。Json和xml的解析我们当时封装了工具类。解析方法网上都有，很简单。

第三方登录：

第三方登录，我的理解就是基于用户在第三方平台上已有的账号和密码来快速完成己方应用的登录或者注册的功能。遵循一个Oauth2.0国际通用协议，允许用户在不提供用户名和密码的情况下，让第三方应用访问一些资源。使用第三方登录时，我们不需要用户再次输入用户名和密码，而是直接通过一个唯一openid来进行授权登录。对于普通用户来说，如果能用QQ、微信、百度、新浪这些平台的账号一键注册登录各个平台，无疑会方便很多。对于我们的应用来说，通过授权，借助QQ、微信这些用户量比较大的第三方平台增强自己的知名度也非常划算。 我们的平台集成了QQ、微信、百度、新浪四种第三方登录方式，实现的方式都是类似的。首先去各大开放平台进行注册成为开发者，并创建应用，填写回调地址，获取appid（应用唯一的识别标志）、appkey（给应用分配的密钥），（名称可能不一样）；下载api文档和sdk开发工具包；就可以开始开发了。 首先在我们网站的登录页面根据api集成第三方登录的logo图标，并给与点击事件，当用户点击此图标时，发送请求，直接跳转到第三方平台的登录页面，第三方平台也会自动检测电脑是否有已登录的账号。登录成功以后，第三方平台会自动调用我们传递的回调地址，并传递回一个code参数；我们拿到code以后，再次调用第三方api提供的接口，传入code、app\_id、appkey等参数，调用获取access\_token的接口(接口调用，有第三方提供的sdk包，直接导入jar包，根据api文档，传递参数调用方法就可以，我们没必要太过关心第三方平台是用webservic接口或httpclient接口。)。获取到access\_token同时，会获取到openid，拿到openid以后，就相当于拿到了登录授权。用openid去自己的用户表中查找是否与对应的用户，如果有，就直接查出用户信息，创建自己的session就可以了。如果没有，则新创建一个用户，把openid放进去。如果还需要其他信息，可以通过openid再次调用第三方平台的接口获取用户信息，如果用户信息还是不够，可以创建完用户以后再次跳转一个页面，让用户不全信息。信息补全以后，创建session，完成登录。这样一个第三方登录就完成了。

## 课程搜索

搜索模块使用的是elasticsearch来完成，ElasticSearch是一个基于Lucene的搜索服务器,他是有一个可视化界面叫head,但是这个head需要布一些环境,就是node.js还有npm（Node.js 平台的默认包管理工具。通过 npm 可以安装、共享、分发代码,管理项目依赖关系。）.我觉得这个东西和vue底层的东西有类似的.之后的话牵扯到就是把数据库的信息导导到索引库里面,我们会用logstarts配置好数据库连接信息,之后放到ElasticSearch配置文件中达到导索引的目的.之后在做的时候也会遇到问题,就是ElasticSearch他不支持跨域的,我们需要在elasticSearch.yml里面加两条命令,设置为true让他允许跨域。另外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。

**面试官：项目中搜索为什么要选择使用Elastic search？**

**面试者：**

Elasticsearch是一个实时的分布式搜索和分析引擎。它可以帮助你用前所未有的速度去处理大规模数据。ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于RESTful web接口。Elasticsearch是用Java开发 的，并作为Apache许可条款下的开放源码发布，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

优点

1.elastcisearch是分布式的。不需要其他组件，分发是实时的。

2.处理多租户不需要特殊配置，而solr则需要更多的高级设置。

3.elasticsearch采用gateway的概念，使得完备份更加简单。

4.各节点组成对等的网络结构，某些节点出现故障时会自动分配其他节点替其进行工作。

**面试官：能说说ES和solr 的差别吗？**

**面试者：**

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。

### Es最小切分和最细切分

切分是基于ik分词器的，IKAnalyzer是一个开源的，基于java语言开发的轻量级的中文分词工具包。采用了特有的“正向迭代最细粒度切分算法“，支持细粒度和最大词长两种切分模式；具有83万字/秒（1600KB/S）的高速处理能力。采用了多子处理器分析模式，支持：英文字母、数字、中文词汇等分词处理，兼容韩文、日文字符优化的词典存储，更小的内存占用。支持用户词典扩展定义

针对Lucene全文检索优化的查询分析器IKQueryParser(作者吐血推荐)；引入简单搜索表达式，采用歧义分析算法优化查询关键字的搜索排列组合，能极大的提高Lucene检索的命中率。

ik 带有两个分词器

ik\_max\_word：会将文本做最细粒度的拆分；尽可能多的拆分出词语

ik\_smart：会做最粗粒度的拆分；已被分出的词语将不会再次被其它词语占有

### Es的正排索引和倒排索引

对于搜索引擎来讲：

正排索引是文档 Id 到文档内容、单词的关联关系。也就是说可以通过 Id获取到文档的内容。

倒排索引是单词到文档 Id 的关联关系。也就是说了一通过单词搜索到文档 Id。

倒排索引的查询流程是：首先根据关键字搜索到对应的文档 Id，然后根据正排索引查询文档 Id 的完整内容，最后返回给用户想要的结果。

单词字典的实现一般采用B+Tree的方式，来保证高效

倒排索引项（Posting）主要包含如下的信息：

1、文档ID，用于获取原始文档的信息

2、单词频率（TF，Term Frequency），记录该单词在该文档中出现的次数，用于后续相关性算分。

3、位置（Position），记录单词在文档中的分词位置（多个），用于做词语搜索。

4、偏移（Offset），记录单词在文档的开始和结束位置，用于高亮显示。

es存储的是一个json的内容，其中包含很多字段，每个字段 都会有自己的倒排索引。

## 课程选择

### 如何进行收藏功能的。

课程收藏的功能描述：用户对一门课程重复的进行收藏与取消收藏操作。点击收藏，将课程收藏成功，并且图标变为已收藏；点击已收藏，则取消对该商品的收藏，并且图标变为  收藏 。这个操作可以重复进行！

下面是我设计此功能的步骤以及关键代码：

        1.首先创建一张收藏表：我的收藏表只有三个字段：id；userID（用户id）；productID（课程id）；

        2.写添加收藏与取消收藏的方法（要根据userID和productID两个参数来实现收藏与取消收藏的）；

        3.我的收藏图标是在课程详情页，所以要在进入课程详情页的时候进行判断，判断该用户有没有收藏该课程！如果该用户已收藏该课程，则图标显示  已收藏，反之，则显示  收藏 。

  进入课程详情页，根据点击图标所显示的文字来判断你所进行的操作，如果点击的文字是收藏，则触发收藏事件，通过ajax进行该用户对该产品的收藏方法；如果点击的文字是已收藏，则触发取消收藏事件，也是通过ajax进行该用户对该产品的取消收藏方法。（注意：这里要先获取到该用户的userID和该商品的productID，这两个参数都可以在进入课程详情的那个方法中获取；“collection.do” 添加收藏的方法，“deleteUcByUPId.do”取消收藏的方法。）

## 圈子交流模块

负责圈子交流模块，通过一对多的问答形式。其中包括根据上级id查询消息，点赞，发布消息等功能。使用spring-data提供的MongoDBTemplate工具类，将信息存入mongodb（减轻数据库压力，实现业务的优化）数据库。

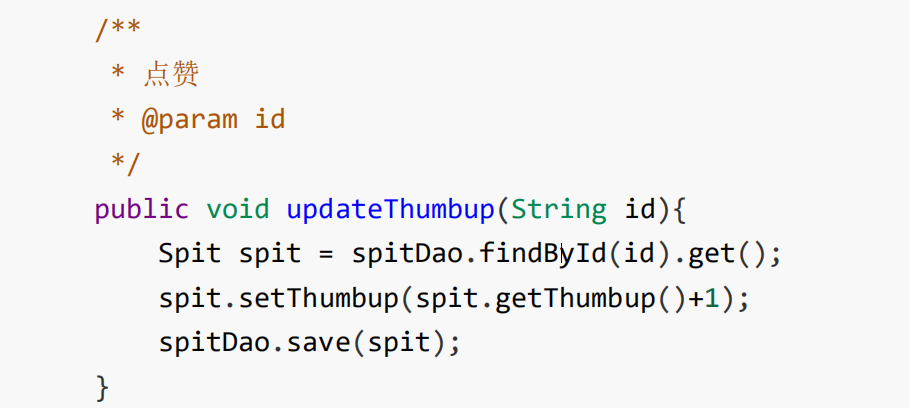
1. 点赞是如何实现的。



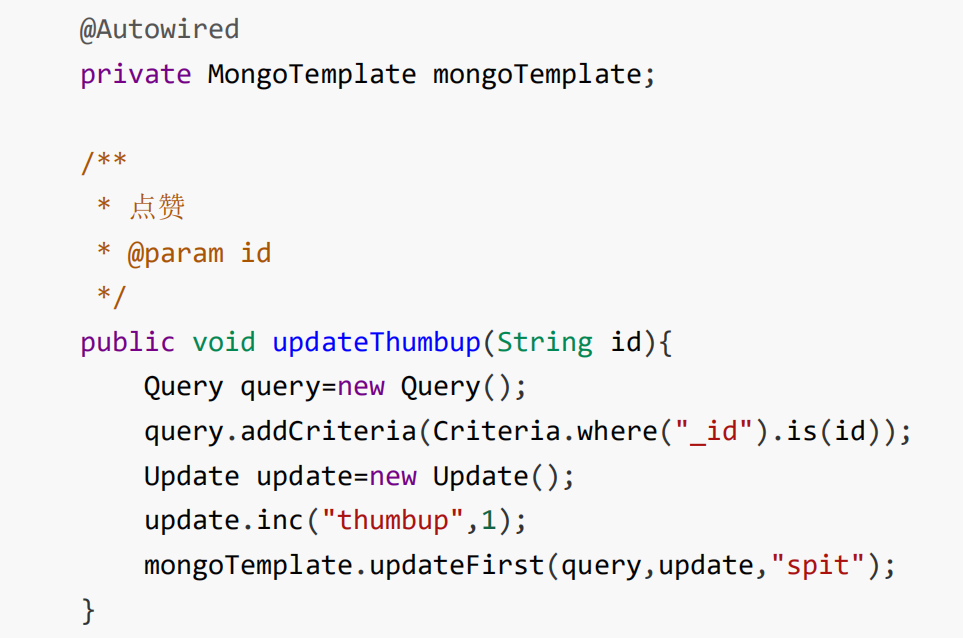
实体类中写上交流模块的字段，service层写上基本的交流增删改查功能

然后我写一个根据上级id查询发布消息的列表（传入本条消息的父级id和分页值）

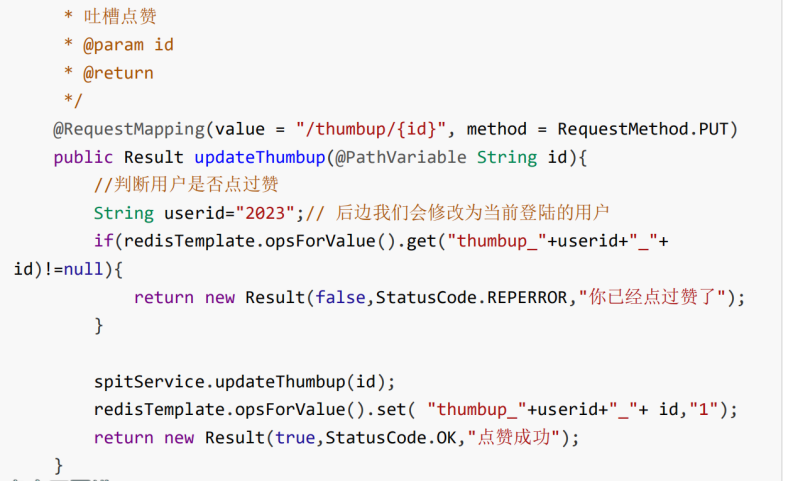
点赞功能我在service层添加一个updateThumbup方法，刚才封装类中有一个点赞数的字段，每当触动了这个方法，点赞数就+1



【以上方法虽然实现起来比较简单，但是执行效率并不高，因为我只需要将点赞数加1就可以了，没必要查询出所有字段修改后再更新所有字段。】所以我们可以使用MongoTemplate类来实现对某列的操作。



然后我们通过redis控制用户不能重复点赞，首先在pom中引入redis的依赖。修改交流服务controller层的业务逻辑。



每一次点赞会进行一次判断，判断本条交流数据是否已经被点赞过了，如果已经被点赞过了，那么就不能进行点赞。

我在做发布评论的时候，对交流服务service层的add方法进行了修改，每当用户发布了一条消息后，会记录 发布日期、点赞数、回复数、状态、分享数。进行判断，如果存在上级id，进行发布评论。

**回复功能开发**

根据id查询评论列表，进行新增与删除的方法（所有的操作是基于mongodb数据库进行操作的）

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

因为这个吐槽、点赞、评论操作，数据量较大，操作的特别频繁，价值很低，当时我们也考虑过用redis，可是我们都知道redis是把数据存储到内存当中的，其缓存的性质远远大于数据存储的性质，并且我们的这些用户的吐槽评论也许对我们来说不重要，但是 对于用户来说是比较重要的， 所以当时我们选择mongodb来进行存储，原因：mongodb是一个跨平台的，面向文档的数据库，支持数据结构非常松散，是类似json的bosn格式，且支持复杂的数据类型，再操作增删改查可以添加很多条件，就像SQL数据库一样灵活

之前用的都是mongodb-driver的一个驱动包，后来就流行了springdatamongodb，吐槽模块采用springdatamongodb这门技术，它是SpringData之一，是操作mongoDB的持久层框架，他的底层已经帮助我们封装好了mongodb-driver，我们在具体操作的时候，只要引入springdatamongodb依赖包能用了

分页操作，需要我们依赖common这个工程，common工程可以直接调用里面的分页等功能，因为我们做的是spring cloud微服务，要涉及到数据之间的传递，需要实现一个serializeable这个接口，继承mongodbrepostry实现类便可进行基本操作

吐槽查询功能，根据父id查询并进行分页，这里分页用的是pageable,这个pageable是个接口，它提供了很多分页的参数，我们需要通过pageRequest.of传分页页数和每页条数就好了

在进行添加吐槽的时候 根据idworker自增一个主键值，再使用这个主键值进行执行添加方法、

我们在吐槽点赞的时候，首先我们判断用户是否点过赞，使用redistemplate调用get方法获取记录，然后进行判断记录是否为空，如果为空的话，说明用户并没有点过赞，我们就使用service调用点赞的方法进行根据用户id进行点赞，并且将记录通过redistemplate调用set方法存储该名用户的点赞记录放到redis当中，如果不为空则返还给我们的前台告知不能重复点赞

Java链接mongodb的时候分5步

1.首先得获得mongodb的链接 使用mongodbClient来选择端口 默认端口是27017

2.选择对应的数据库 通过mongodbClient调用getdatabase这个方法选择自己的数据库名称

3.根据数据库调用getcollection方法选择对应的集合

4.如若是查询集合内的所有文档（一个文档代表一条数据）直接使用find方法查询后，for循环遍历输出便可查询全部

5.打开mongodb这个链接后，我们也需要关闭这个链接mongodbClient调用close即可关闭

如果我们要进行多条件查询的时候，我们可以用 new BasicDBObject来构建查询条件，这里面存放的都是键值对，如果是多条件的话，就可以再new BasicDBObject中再嵌套一个实 现多条件查询

添加操作直接使用insertOne即可添加，由于我们这个mongodb里面存放的都是bosn格式，所以再添加的时候，直接new hashmap，在map里通过put方法添加好数据后，用document对象可在调用insertone方法进行添加

你的工程是如何操作MongoDB的？

在之前项目中都是引入mongodb-driver

这个驱动包的，后期用的是springdatamongodb

在项目的哪些场景下使用MongoDB ?

吐槽 、文章评论

为什么在吐槽和文章评论中使用Mongodb而不使 用mysql?

吐槽和评论都是数据量较大且价值较低的数据，为了减轻mysql的压力，我们使用 mongodb

# 食路通———————————————

## 单点登录

1.把登录功能单独拿出来创建一个web项目这个web项目就是SSO(Singlon sign on)单点登录功能，提出一个登录页面  
  
当我们在页面涉及到与下订单或者购买的时候会有一个拦截器来判断你有没有登录，如果没有登录的话就会给你返回到SSO里面的登录页面  
单点登录解决方案：

首先，我们要明白什么是单点登录，单点登录，英文简称SSO，

因为这个商城，主要是一个分布式的架构，分布式的架构在我们web端,有多个web项目组成，多个web项目组成就会有这样的一个问题，系统的访问是需要权限的，那么就是

说假设它在其中某个项目登录的话，我们以往是把登录人的信息存入到session

里面，但是分布式项目当中，你存到某一个项目的session当中，其他的项目是

访问不到的，这个时候呢，就会设计到一个session共享的一个方案，

我们这里选用了一个单点登录的技术呢，是一个开源的框架CAS。

CAS是耶鲁大学发起的一个开源的项目。它主要的目的就是为web系统提供一个

通用可靠的单点登录的方法。是一个企业级的单点登录解决方案。安全这一块

也做的非常好，CAS的工作原理：

CAS主要有两个端，CASclient和 CASserver端。

当我们分布式的项目中呢会有很多个节点，我们这每个节点会给它配置一CASclient客户端，我们进行每个节点访问的时候呢，CASclient它提供了对应的过滤器，过滤器会检查你当前的登录状态，其实就是在检查你登录的时候，有没有一个凭证，这个凭证呢，咱们一般用cookie来存储的，发现你没有凭证之后，他这个过滤器会把你重定向到CASserver端，也就是说给你重定向到CAS统一认证的那个界面，CASserver呢提供一个登录的页面，还有对应的登录方法和登出的方法和登录成功的页面，这个页面呢，我们在项目中呢会使用我们自己的登录页面，要更改CAS默认提供的登录页面，CASserver就是别人给你写好部署好的一个war包,war主要就是运行在Tomcat下面，用Tomcat将它跑起来，然后呢，刚刚提到，当我们进行访问的时候，假设访问成功了，认证中心会再重定向到我们访问的节点上，并且给我们发放一个票据，下一次你再访问其他应用的时候呢，你会带着这个票据，过滤器会判断你这个票据是否正确，他会拿着你这个票据重定向认证中心去认证查看你的票据是否有效，如果对了，你这个节点也可以访问在我们做的这个项目中呢，我们主要将securityzx 和CAS结合到一起使用的，security主要实现的就是系统登录授权。然后结合CAS实现一个单点登录

具体使用 ：先部署一个CASserver端 去CAS的官网上下载，复制war，在这个项目里呢，我们使用的是4.0.0的,有了这个war包之后呢，我们将这个war包放到Tomcat下面解压这个war包，之后我们就可以访问了。访问的默认地址：8080/war的名字/login页面，就可以进行一个CAS的亮灯了操作首先根据我们的项目来讲。我们要修改Tomcat的端口。sever.xml。用这个CAS得时候，我们修改登录页面，前面也提到过，我们要改成我们项目中自己的登录页面，登录页面主要在web的文件夹下，有个defult.ui的文件下，将我们项目中的静态页面放进去，替换他原来的login.view这样的一个JSp。

改好这个登录页面之后，我们还要改一个CAS的 国际化配置，因为这个CAS支持多语言的，默认配置的语言是英文，我们需要将它改成中文，配置好之后，我们开始项目的整合。我们将这个CASserver跑起来之后呢，

其他的web项目需要配置什么呢？需要引入CASclient的jar包，接着呢，我们需要配置把他和security配置到一起，因为前面我们说了，我们使用CAS呢，主要使用security和CAS进行的一个整合，security是spring出的一个安全jar包，它本身也提供了security和CAS 的整合方案，整合方案之后呢 ，我们只需要更改，security的配置就可以了

security主要就是要配置，当没有权限的时候，要跳转到登录页面，主要配置

就是拦截之后，登录页面的一个地址，还有登录成功之后，返回的一个地址，还有security本身的配置。要配置要拦截的哪些东西，和登出的地址

**利用redis实现：**我们的单点登录系统，主要包含了登录验证、token校验、注销、注册几大功能，单点登录系统提供了统一的登录

注册页面，提供了统一的登录token校验接口。 单点登录的主要原理就是在登录成功以后，生成一个令牌，这个令牌要求每次登录

一不可重复，我们就简单的用了一个随机的UUID，因为我们的系统在部署时，各个模块都是通过Nginx映射到同一个一级域名下

cookie只要把他的作用域设置成一级域名，就可以在所有同一个一级域名下的模块中共享。所以我们把随机生成的token，以字

“token”为key，放在cookie里边，然后用生成的token做key，用户对象信息转成json字符串后，作为value，放到redis里

都设置有效期30分钟； 然后做了一个统一校验token的接口；各个模块在自己的拦截器里边，调用此token校验接口，验证是录，

如果返回null，则说明没有登录，拦截到统一的登录页面，并把进入拦截到的url放入cookie里边，方便登录成功以后，获取

这个url，进行原路径跳转，而不是每次登录都进入首页，提供用户的体验度。如果返回用户信息，则说明已经登录，模块创建自己的

session，并放行url。统一校验token的接口的主要流程是，首先从cookie里边获取到token，然后通过token到redis里边获取用

户信息。这两个过程，任何一个失败，都直接返回null，如果成功，就把cookie和token的值从新设置一遍(这个是为了刷新有效期)；

这样就实现了多个模块只需要登录一次就可以的流程。 还有就是注销，注销也是调用统一的注销接口，注销时需要首先从cookie中

获取token，根据token删除redis中的用户信息，然后在删除cookie中的token。

**cas实现：**1、 用户访问网站，第一次来，重定向到 CAS Server，发现没有cookie，所以再重定向到CAS Server端的登录页面，并

且URL带有网站地址，便于认证成功后跳转，形如 http ://cas-server:8100/login?service=http ://localhost:8081

注意：service后面这个地址就是登录成功后要重定向的下游系统URL。

2、 在登陆页面输入用户名密码认证，认证成功后cas-server生成TGT，再用TGT生成一个TGC（存到浏览器cookie中的）和ST（令牌，只能用一次）。 然后再第三次重定向并返回ST和cookie(TGC)到浏览器

3、浏览器带着ST再访问想要访问的地址http ://localhost:8081/?ticket=ST-25939-sqbDVZcuSvrvBC6MQlg5（注意：ticket后面那一串就是ST）

4、浏览器的服务器收到ST后再去cas-server验证一下是否为自己签发的，验证通过后就会显示页面信息，也就是重定向到第1步

service后面的那个URL首次登陆完毕。

5、再登陆另一个接入CAS的网站，重定向到CAS Server，server判断是第一次来（但是此时有TGC，也就是cookie，所以不用去登

陆页面了），但此时没有ST，去cas-server申请一个于是重定向到cas-server，形如：

http: //cas-server:8100/login?service=http ://localhost:8082 && TGC(cookie) (传目标地址和cookie）

6、cas-server生成了ST后重定向给浏览器http ://localhost:8082/?ticket=ST-25939-sqfsafgefesaedswqqw5-xxxx

7、浏览器的服务器收到ST后再去cas-server验证一下是否为自己签发的，验证通过后就会显示页面信息（同第4步）

## Solr搜索

对于搜索我们采用的是solr搜索引擎，我们为搜索启了一个单独的服务。

Solr是如何做的呢？

使用前进行下solr的war包，将war部署到服务器中去。

Solr默认的分词对中文不是很友好，中华文化，博大精深嘛。所以需要进行配置ik分词器，ik是一个开源的基于java语言开发的轻量级的中文分词工具包。

分词的配置：将分词的jar包拷贝到solr/web-inf/lib下

然后再solr/web\_inf/下创建一个文件夹 ，名为classes，在这个文件夹中，我们放扩展词典和扩展停止词典，

接着创建存放solr数据的文件夹，solrhome，将这个solrhone的地址，在solr的web.xml中配置一下。

Schema.xml也是在solrhome里的，这个schema.xml主要配置的就是solr数据库中的字段中文分词器也是在这里配置的 配置一个fieldtype 让solr可以使用中文分词器

进行域的配置：

普通域：就相当于实体中的属性。

复制域：将某个field中的数据复制到另一个域中，主要用在我要进行搜索的时候，可以更大范围的搜索出商品，提高为用户展示的信息量。

动态域：动态域使用了通配符，主要是为了批量的向索引库添加索引

这些关于solr配置完毕之后，我们就可以结合项目开发了

首先，我们要将数据库里面的数据，一次性的导入到solr里面。

那我们在做sol导入数据，有两种方式，一种是导入插件，但是这种插件，只能做一些简单的业务逻辑，所以我们在这个项目当中呢，是自己用代码进行导入的，先从数据库中将数据查询出来，在pom.xml中配置springDataSolr依赖，然后利用solrtemplate来将数据导入到solr里面。

要想实现高亮，要在那个字段高亮，怎么去亮。我们自己设置一个标签，例如:<em style=“color:red”>属性</em>

先创建一个高亮的设置选项，设置到那个域来高亮怎么高亮，设置样式，要在这个属性前边加个前缀，在属性后边价格后缀，

在哪个域上查询这个效果，进行封装，设置完之后就可以根据这个条件进行查询。得到处理过的数据列表之后，高亮并没有直接显示，我们高亮的字符串取出来在设置。进行循环。

而我们在做高亮的时候，显示的html代码会原样输出，这是angulars为了防止html攻击采取的安全机制，我们如何显示html的结果呢，我们用到了$sce服务的trustAsHtml方法来进行转换。（过滤器）

而我们在做用关键字来查询列表的时候，

当用户输入关键字搜索后，除了显示列表结果外，还应该显示通过这个关键字搜索到的记录都有哪些商品分类。

根据第一个商品分类查询对应的模板，根据模板查询出品牌列表

根据第一个商品分类查询对应的模板，根据模板查询出规格列表

当用户点击搜索面板的商品分类时，显示按照这个关键字查询结果的基础上，筛选此分类的结果。

当用户点击搜索面板的品牌时，显示在以上结果的基础上，筛选此品牌的结果

当用户点击搜索面板的规格时，显示在以上结果的基础上，筛选此规格的结果

当用户点击价格区间时，显示在以上结果的基础上，按价格进行筛选的结果

当用户点击搜索面板的相应条件时，隐藏已点击的条件。

首先我们在使用关键字之后怎么根据关键词查询出分类表。

搜索面板的商品分类需要使用Spring Data Solr的分组查询来实现

为了能够提高查询速度，我们需要把查询面板的品牌、规格数据提前放入redis

查询条件的构建、面板的隐藏需要使用angularJS来实现

后端的分类、品牌、规格、价格区间查询需要使用过滤查询来实现

首先我们商品分类是根据关键字的出来的，查出几种就是几种，我们用到了Solr里的Spring Data Solr 里的一个group选项，可以设置以哪一个域作为搜索条件，根据结果组装程一个集合展示出来。

而我们做品牌和规格的时候，我们存几种数据，通过商品分类找到模板id，

将商品分类数据、品牌数据、和规格数据都放入Redis存储。

当用户进入运营商后台的商品分类页面时，将商品分类数据放入缓存（Hash）。以分类名称作为key ,以模板ID作为值

当用户进入运营商后台的模板管理页面时，分别将品牌数据和规格数据放入缓存（Hash）。以模板ID作为key,以品牌列表和规格列表作为值

**咱们这个项目是一个大型的电商项目，商品数据非常多，在用户访问商品详情的时候，由于每款商品都不一样，每款商品的详情页都是动态生成的，每查看一次，页面就要重新渲染一次，渲染的过程是消耗时间的，当用户访问量很大的时候，在一定程度上降低了系统的性能，所以在这一块，我们要使用网页静态化来帮我们提升系统的性能，目前实现网页静态化的技术有好几种，其中有volocity和freemaker，那在这个项目中呢，我们使用的是freemaker这个技术来实现网页静态化**

在商家在添加完商品的之后，运营商会对这个商品进行审核，

当审核通过之后呢，我们就会对这个商品进行一个商品详情页生成的这么一个过程，

就是先生成一个商品模板，通过商品的spu信息去数据库进行查找，查找到之后把

商品信息通过IO流写入到到生模板中，生成一个网页文件，然后再当用户查看这个

商品详情页面的时候，那他访问的就是静态页面了，这个页面里面包含了商品的分类啊，

图片啊，商品的介绍啊，规格啊，这些sku的信息，但是我们在这块儿的工作不是运营商来完成的，

在这我们用到了一个消息中间件 消息中间件利用高效可靠的消息传递机制进行平台无关的数据交流，并基于数据通信来进行[分布式系统](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F" \t "https://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E4%B8%AD%E9%97%B4%E4%BB%B6/_blank)的集成，消息中间件产品主要有activeMq，RabbitMQ（金融方面的），ZeroMQ（史上最快的消息队列系统）Kafka是用于海量数据，特点：高吞吐，在一台普通的服务器上既可以达到10W/s的吞吐速率 ，activeMQ，他有两种模式，一个是点对点，还一个是发布订阅，在这我们用的是发布订阅模式，为什么是订阅模式呢？因为我们后期要部署到nginx上。

具体生成 详情页是一个服务，在这个服务上有一个pom依赖，依赖了freemarker的相关依赖

因为用的是发布订阅模式，他有一个消息监听接口会接受到传来的spu的id，

接到这个id之后根据这个id把它对应的信息查询出来，查完之后在服务中内置一个商品详情页的模板，

用freemarker把需要变得地方用变量替换掉，然后生成静态网页

## 详情页

在spring配置文件里配置一个freemarkerconfig的bean，这个bean是spring提供的对freemarker支持的配置文件，

里面提供了两个属性，一个是模板的目录，一个是字符集，有了这个配置，就能获取到配置对象，就不用每次设置目录和字符集

**生成文件的步骤：**

使用步骤：

第一步：创建一个 Configuration 对象，直接 new 一个对象。构造方法的参数就是 freemarker的版本号。

第二步：设置模板文件所在的路径。

第三步：设置模板文件使用的字符集。一般就是 utf-8.

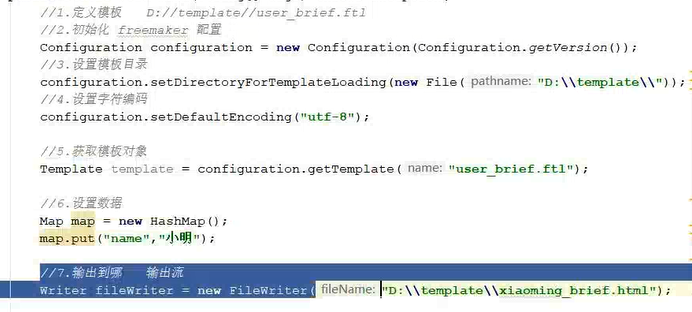
第四步：加载一个模板，创建一个模板对象。 //模板

第五步：创建一个模板使用的数据集，可以是 pojo 也可以是 map。一般是 Map。 //数据

第六步：创建一个 Writer 对象，一般创建一 FileWriter 对象，指定生成的文件名。

第七步：调用模板对象的 process 方法输出文件。

第八步：关闭流



我在我们这个项目里负责的是商品详情页这个模块，用户在点击商品详情的时候，会跳转到商品详情页，

我们商品详情页的页面就是一个sku的信息，所以我们在做商品详情页的时候。

我们在配置这个的时候，我们要配置一下生成的模板，所以我们要在配置文件中配置freemarkerConfig这个对象，里边有两个属性，一个是配置这个模板路径在哪里，还有一个是字符集编码。我们在做的测试的时候，我们首先把它生成在某个盘当中，当我们在做生产环境的时候，我们会部署一个nginx，把nginx文件读取的目录设置到装静态页的那个盘。



我们在服务层做的时候首先是加载模板，加载模板之后我们把数据放在模板当中，然后看一个输出到哪！

输出的的地方是生成模板的路径+sku商品的id。

## 优惠券

优惠券领取也是我负责的，领用优惠券功能用redis数据库进行存储，首先查询redis缓存，如果缓存存在，返回结果；缓存不存在则查询数据库。进行争夺分布式锁，成功获得锁后，再次判断缓存是否存才，如果缓存不存在，把查询数据库的结果循环放入缓存。最后进行释放分布式锁。