**十次方项目**

**介绍**

《十次方》是程序员的专属社交平台，包括头条、问答、活动、交友、吐槽、招聘六大

频道。

**系统架构**

《十次方》采用前后端分离的系统架构

后端架构为 SpringBoot+SpringCloud+SpringMVC+SpringData  我们把这种架构也称之为全家桶

前端架构：Node.js+Vue.js+ElementUI+NUXT

**使用的技术**

我们主要做后端,我们使用到了 springcloud 的集中配置中心,负载均衡Ribbon,熔断器springcloudHyst, Spring Cloud Zuul,信息总线Spring Cloud,SpringBoot,JWT.其中也用到了一些持久化技术,SpringDataJPA,SpringDataRedi,SpringDataMongoD,SpringDataElasticse.

前端element UI库,node.js,npm,axios,Mock.js,VScone,Vue.js,基于NUXT,Es6,easy Mock,Swagger,webpack

数据库索引库消息队列,技术栈,,mycat数据分片,Redis集群,mongoDB集群,Mysql读写分离,Elasticsearch集群,RabbitMQ

**模块介绍**

我们的十次方工程共分为18个子模块（其中17个是微服务）

因为有一个父工程,然后17个子模块中,common是工工资模块,用来提供共有的依赖,base模块是用来提供基础的微服务.

**表结构**

我们这里采用的分库分表设计，每个业务模块为1个独立的数据库。

登录业务

用户模块中,同样涉及到了两个小模块,一个是用户模块,即普通用户,另一个是管理员模块,即管理者,用户模块涉及到了,对用户的一系列CRUD操作,还有登录,注册,发送手机验证码,关注用户,取消关注 ,查询用户的粉丝和关注,,还有ID列表.管理员模块中,仅涉及到了CRUD操作和登录.

触发登录条件

当用户点击注册登录按钮时会触发注册登录流程或者用户在未登录情况下在头条板块进行发布、点赞、收藏操作；在问答板块进行回答、点赞、提问、评论、收藏、添加标签操作；在活动版块点击我想参加、立即报名、评论操作；在交友板块发布邀约、点击喜欢、不喜欢、私信按钮、完善信息操作；在吐槽板块评论、点赞、收藏操作；在招聘页面收藏、投简历操作 以上几个操作会触发注册登录条件。

**交友模块**

交友模块中,涉及了添加好友或者非好友,删除好友的功能,.

特性描述：喜欢：用户点击喜欢后在对方列表中优先推荐，如对方也喜欢后则在通知栏进行通知，并可进行私信 不喜欢：当用户点击不喜欢后，对方讲被放入待选池中，在7天后将会重新被推荐。喜欢：用户每喜欢一个人则在次数显示一个喜欢数 被喜欢：改用户被其他用户喜欢数量。

**活动模块**

活动模块中,涉及了增加活动,获取活动列表,根据活动ID查询活动,修改活动,根据ID删除活动, 根据条件查询活动,活动的分页展示,根据城市来展示分页内容

招聘业务：热门企业、最新职业、热门企业

招聘技术：采用SpringDataJPA、SpringDataRedis

特性描述：热门搜索：根据平台用户搜索习惯，显示排名前5的关键词工作城市：根据平台中存在的公司

**问答模块**

搜索业务 浏览问题 发布问题 回答问题 关注

问答模块中,涉及了两个小模块,一个是问题,另一个是回答.在问题模块中,涉及到了,增加问题,问题列表,根据ID查询,修改问题,根据ID删除,根据条件查询问题,问题的分页展示,最新问题展示,热门回答展示,等待回答的展示.在回答模块中,涉及到了增加回答,回答列表,根据ID查询回答,修改Reply,,根据ID删除,回答分页,根据问题ID查询回答列表,回答问题,我的回答列表.

**吐槽模块**

也是涉及了CRUD操作,还有吐槽点赞,根据条件查询和分页,以及根据父级ID查询.

其他的一些模块不涉及业务,不再做过多介绍

**搜索业务**

文章搜索，同步数据库

搜索技术：SpringDataElistciSearch,IK,Elasticsearch

**负责模块**

我主要负责的是吐槽 使用SpringDataMongoDB完成吐槽微服务的开发

吐槽和评论两项功能存在以下特点：数据量大,写入操作频繁, 价值较低

对于这样的数据，我们更适合使用MongoDB来实现数据的存储

因为吐槽模块的数据量太大,且在当前场景中,丢失一两条并无所谓,所以说,采用MongoDB数据库吞吐量大,比关系型数据库要大得多,所以数据再多也没多大压力.

当用户发布吐槽时,就往数据库里面添加一条,列表展示就是查询吐槽的数据库,

剩下的修改和删除也不在解释,流程是差不多的.

下面说一下对吐槽的条件查询,因为用户在浏览的时候,可能想要对应浏览相关的内容,这时候就需要条件查询来找到对应的内容,

另外,根据吐槽我们还可以进行”吐槽”,其实就是相当于吐槽的评论,但是它是根据已有的吐槽进行的,所以,这就相当于它的子,我们可以根据父\_id来查询对应的子,

我们也可以进行点赞操作,即设置一个字段来记录点赞个数,然后用我们可以通过redis控制用户不能重复点赞