# 十次方

## 一:表述与架构

十次方是一个交友平台,供用户体验对好友的一些操作,查看文章的功能等.

后台的每一个功能都是一个微服务,,将搭建好的服务功能制作成镜像,把镜像当做容器,通过springcloud来实现服务之间的通信,前端通过调接口,来实现功能.

## 二.框架

前后端分离的系统架构,后端采用sprinboot+springcloud+springmvc+springdata 全家桶的,使用springdata做持久层框架,优点就是:不管是否是关系型数据库,操作几乎一样,,,使用全家桶的框架,开发速度快,可以把精力集中到业务上.数据库用mysql和Mongodb,

Mongodb:

在本项目中,将吐槽模块的数据,存放到mongodb中,其他模块数据存放在mysql中,

Mongodb使用场景:海量数据,读写操作频繁,并且该数据不重要的场景

Mongodb是属于非关系型数据库,存放的数据格式是bson,和json很相似,

支持的基本数据类型有null,布尔,字符串,日期,数值,都以bson格式存储.

## 三.技术

1.Restfull设计风格,7个http请求方式:post,get,put,delete,path,head,options

在项目中用到: post,get,put,delete,

安全:不会出现脏读,不可重复读,幻读,就是安全的

幂等:在操作成功的前提下,不会对数据库造成不好的影响

安全:get (其他三个不安全)

不幂等:post (其他三个幂等)

2.rabbitMq :

消息中间件.

3.ElasticSearch:

Logstash同步数据,

5.认证机制:JWT(token)

## 四.模块

分18个子模块:17是微服务

基础模块,用户模块,问答模块,吐槽模块,交友模块,活动模块,招聘模块,文章模块..

## 五,模块详情:

### 文章模块

**业务流程:**

将数据库中的数据通过logstash与索引库进行同步,在同步时,需要涉及到索引,ik分词,的操作,,然后在搜索框中输入搜索数据,在索引库中进行查询

**技术: ElasticSearch,**

**在属性上注入@Field注解有三个表示:**

1. 是否索引(就是看该域是否被搜索)
2. 是否分词(就在搜索时,是案整体匹配还是单词匹配)
3. 是否存储(是否在页面上显示)

**搜索分为两大类**:

1.搜索引擎(百度,谷歌),,

2.站内搜索(本公司数据库中的数据)

**ElasticSearch**,:倒排索引(通过值来找key)

**数据格式上的区别(和solr的区别):**

Slor:JSON,XML,字符串

ES:JOSN,接口风格支持restfull

**集群**:

Slor:依赖于第三方

ES:靠自身实现

**Logstash**:同步时,索引库的数据并不会删除

### 用户模块

**业务流程**: 用户分为普通用户和管理员用户,

普通用户可以进行登录,注册

管理员用户不可以自己注册,需要分配特殊管理权限才能进行登录.

普通用户可以注册,发送验证,然后将生成的验证码一份保存到redis当中,用来与用户输入的验证码进行匹配,如果匹配成功,那么就进行下一步,

在发送短信验证码时用到消息队列,rabbitMq,通过阿里云通信来发送

**技术:**

**RabbitMq**:消息中间插件;

和activemq,kafka的区别

Active,kafka,rabbit都是安全的,从安全级别由高到低来排序: Active,kafka,rabbit

Kafka使用场景是大数据,速度快

Rabbitmq是经过交换器

Rabbitmq有三种发送模式:

直接模式:默认经过空字符串交换器

分裂模式:fanout: 缺点,用交换器就写死了

主题模式:生产者定义三种形式:

Good.#

#.log

Good.log

**JWT**(JSON WEB Token)

发送验证码: 引用commons-langs,生成随机数,保存一份在redis缓存中,

注册,验证码:JWT token,网关加密.采用Bcrypt算法加密

有三部分组成: 1.头部,base64编码(默认的)加密方法

签名算法HS256

2.载荷:,主要的信息(包含自定义的)

3.头部+载荷+盐+密

**Token优点**:

支持跨域(cookie不支持)

无状态

去耦

更适用于移动应用(cookie只在浏览器应用)

无状态:不需要从本地查(效率快)

有状态:需要从存放的地方查(效率慢)

### 交友模块

**业务流程:**

当用户向一分用户点击喜欢,那就会将这条数据添加到好友表,但状态为0,表会不是互相喜欢,如果被喜欢的用户,也向此用户点击喜欢,那么该用户的数据回添加到好友表,并且此时的两个用户状态是1,表示互相喜欢,

也可以进行不喜欢操作,会执行两步,一步是,从表中删除数据,第二步,对方的状态为改为0,

通过eureka服务之间的调用,来实现交友模块中的添加好友,非好友,更新关注数和粉丝数,以及删除好友.

**技术:**

**eureka**: 服务注册于发现,他和zk很相似,从服务注册中心上来讲,eureka要比zk更专业,因为服务中心要保证的是服务的可用性,eureka保证的就是ap,可用性和分区容错性,而zk保证的是cp,一致性和分区容错性.zk如果因网络出现故障,整个服务会瘫痪,而eureka会切换到其他节点上,不会影响可用性.

Feign:实现服务间的调用

通过 @EnableFeignClients注解来表名当前类调用了其他的服务

@FeignClient("…")表示从哪个服务中调用的功能

此项目中,需要搭建一个服务器,让其他服务发现.注册到eurekaServer中.其他服务就是eukeka客户端,

**熔断器(Hystrix(实现熔断器)):** 当一个分支出现异常,去执行另一分支.要自定义一个熔断器实现类,实现接口,修改实现接口的注解@FeignClient(,fallback=实现类)

**雪崩效应;**如果微服务架构中存在多个服务之间的调用,其中的一个提供者服务出现故障,会连其他服务也不可用,并将不可用逐渐放大,就像一个雪球一样,越滚越大,就形成了雪崩效应**.**

**网关(Zuul):**

**可解决的问题:**1.当客户端多次请求不同的微服务时,会增加客户端的复杂性

2.认证复杂,每个微服务都需要独立认证

3.难以重构,如果客户端直接和微服务系统通信,需要重新划分微服务是,难以实施

**Zuul的核心组件:是一系列的过滤器,对每一个资源要求验证,拒绝不符合的请求.**

**当前台在去调用后台微服务接口时,需要通关网关的认证,再由网关进行对后台微服务发送请求.**

***SpringCloudBus(消息总线组件):***当我们修改码云中的配置文件时,不需要重新启动程序,向 mq发送一个请求,可以实现不重启程序的情况下,就可以更新数据

***自定义配置:***@RefreshScope(用于刷新配置)