

CS307Project2

Information

Lab03-4-34

汤玉磊 12111908

张未硕 12111905

Group Number: 307

Contribution

成员	贡献比	建表	基础部分	高级部分
汤玉磊	50	修改	前端	前端
张未硕	50	绘制	后端	后端

0.技术栈

前端

VUE

- 网络请求 Axios
- 界面美化 Vanta.js Element UI
- 路由管理 Vue Router

后端

架构

- Springboot
 - 数据持久化 MybatisPlus

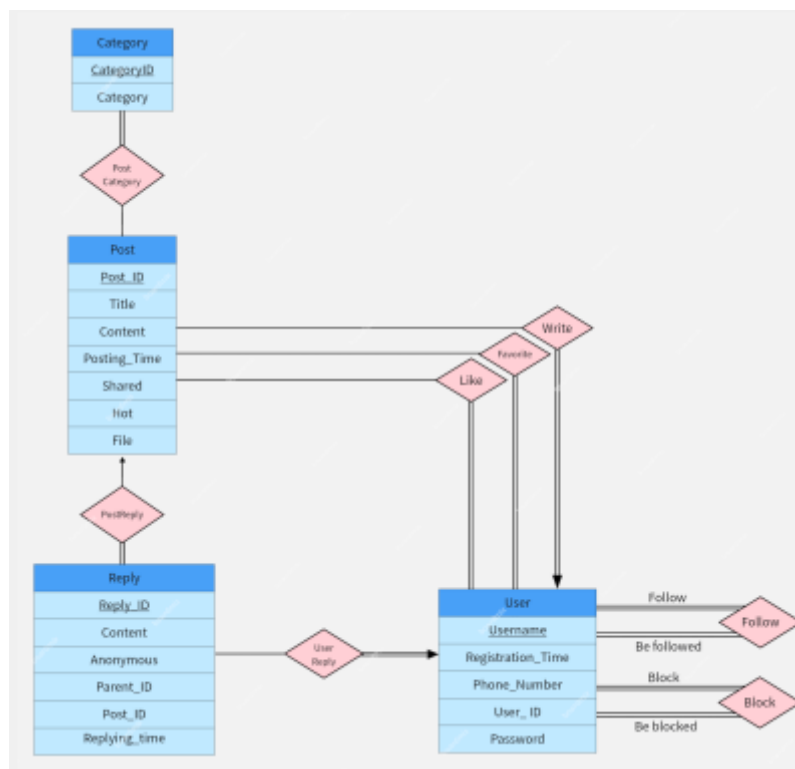
数据库

- Opengauss/PostgreSQL

1. 基础设计

ER图

本次project中，我们绘制的ER图如下 绘制软件：boardmix, <https://boardmix.cn/>



强实体集（4张）

Users

属性	类型	含义	约束
username	varchar(50)	每个user的用户名	primary key
phone_number	varchar(50)	手机号	unique, not null
registration_time	timestamp	注册时间	not null
user_id	VARCHAR(20)	每个用户独一无二的id	not null
password	VARCHAR(20)	用户的密码	not null

Posts

属性	类型	含义	约束
post_id	bigint	每一个post唯一确定的id	primary key
content	varchar(1000)	post的内容	not null
title	varchar(100)	post的标题	not null
posting_time	timestamp	发帖时间	not null
shared	bigint	用于储存帖子的转发信息，为0表示帖子为原创，否则为转发原帖的id	not null
hot	bigint	用于储存帖子的热度信息	not null
file	varchar(200)	用于储存帖子所带文件的地址	

Replies

属性	类型	含义	约束
reply_id	BIGINT	每一个reply唯一确定的id	primary key
content	VARCHAR(1000)	replyd的内容	not null
stars	BIGINT	reply的点赞数	
anonymous	Boolean	用于表示该回复是否为匿名发出	not null
post_id	BIGINT	用于表示该回复所在的帖子id	not null
parent_id	BIGINT	若非空则代表该reply回复的reply的id	

Category

属性	类型	含义	约束
category_id	BIGINT	每一个category唯一确定的id	primary key
category	VARCHAR(30)	category的内容	not null

关系表（9张）

UserReply

表Users一对多于表Reply

属性	类型	含义	约束
username	varchar(50)	来自表Users的主键username	foreign key, composite primary key
reply_id	bigint	来自表Reply的主键reply_id	foreign key, composite primary key

UserWritePost

表Users一对多于表Post

属性	类型	含义	约束
username	varchar(50)	来自表Users的主键username	foreign key, composite primary key
post_id	bigint	来自表Post的主键post_id	foreign key, composite primary key

UserLikePost

表Users一对多于表Post

属性	类型	含义	约束
username	varchar(50)	来自表Users的主键username	foreign key, composite primary key
post_id	bigint	来自表Post的主键post_id	foreign key, composite primary key

UserFavoritePost

表Users一对多于表Post

属性	类型	含义	约束
username	varchar(50)	来自表Users的主键username	foreign key, composite primary key
post_id	bigint	来自表Post的主键post_id	foreign key, composite primary key

PostCategory

表Post一对多于表Category

属性	类型	含义	约束
post_id	bigint	来自表Post的主键post_id	foreign key, composite primary key
category_id	bigint	来自表Category的主键category_id	foreign key, composite primary key

PostReply

表Post一对多于表Reply

属性	类型	含义	约束
post_id	bigint	来自表Post的主键post_id	foreign key, composite primary key
reply_id	bigint	来自表reply的主键reply_id	foreign key, composite primary key

UserFollowUser

表Users多对多于表Users

属性	类型	含义	约束
user_follower	varchar(50)	来自表Users的主键username 表示关注者的用户名	foreign key, composite primary key
user_be_followed	varchar(50)	来自表Users的主键username 表示被关注者的id	foreign key, composite primary key

UserBlockUser

表Users多对多于表Users

属性	类型	含义	约束
user_blocker	varchar(50)	来自表Users的主键username 表示屏蔽者的用户名	foreign key, composite primary key
user_be_blocked	varchar(50)	来自表Users的主键username 表示被屏蔽者的id	foreign key, composite primary key

UserLikeReply

表Users一对多于表Replies

属性	类型	含义	约束
username	varchar(50)	来自表Users的主键 username	foreign key, composite primary key
reply_id	bigint	来自表Reply的主键 reply_id	foreign key, composite primary key

Advanced Requirements

Open Guass数据库

安装

1. 利用VMware安装Centos7.6系统
2. 在Centos系统上安装Docker镜像，从DockerHub拉取Opengauss镜像
3. 利用Docker创建该镜像的实例，自此安装完毕

使用

该数据库是基于PostgreSQL开发的，所以使用方法和PostgreSQL一致，需要把数据库代码中的FUNCTION改为PROCEDURE即可正常使用

完善API

匿名发言

我们的匿名发言使用的是假匿名，即在服务器中记录了匿名发言人的具体信息，而在前端显示时则会将其具体的名字改成“匿名”（见下图）。



具体实现方法是在Reply表中添加一个布尔型的**anonymous属性**，如果用户勾选匿名，则该属性为true，反之则为false，在前端，我们对于anonymous为true的会将其显示为“匿名”，否则会显示用户的名字

图片/视频播放

在该部分镶嵌在发帖页面中，我们在这里使用 `POST /post/uploadPic/{username}` 请求来上传图片，在服务器端，会将上传的图片放在每个用户对应的文件夹内。在数据库中，我们在**posts**表中添加了file属性用来存储，文件在服务器中**静态资源**的相对路径，例如，用户tang上传了一张1.png的图片，则在该post中对应的file属性会填写/tang/1.png。

在前端帖子中会使用默认组件`<video>`和``搭配`v-if`和`v-else`来调用显示该视频/路径，访问资源的路径为后端Springboot静态资源地址（<http://localhost:8088/staitc>）加上后端返回的数据表**posts**中的相对路径**file**。

屏蔽

我们在前端中，增加了“屏蔽”按钮，如果用户点击了该按钮，则前端会向后端发送 `POST /post/userLikePost/${this.id}` 请求，同时将按钮切换为“取消屏蔽”，在屏蔽后前端在**帖子列表**中将会不会显示有关屏蔽用户的一切帖子。

具体实现方法是在分页返回post的时候，筛选`select * from posts`的时候使用该语句 `NOT IN (SELECT user_be_blocked FROM UserBlockUser WHERE user_blocker = #{username})`来筛选屏蔽列表中用户的帖子。

热搜榜

在热搜榜中，我们在**posts**表中定义了一个新属性hot，该值等于被点赞+被评论*3+楼中楼+收藏*2+被分享*2，在每次执行点赞，评论，楼中楼，收藏，被分享操作时，都会触发该对该hot的修改。在获取热度榜时，我们只需要在select语句后增加 `order by hot desc` 即可

多参数搜索

我们的多参数搜索类似知网高级搜索的方式，支持自定义添加多个条件（见下图），各个条件之间是AND的关系，允许重复的条件，我们支持的参数类型有**标题**，**内容**，**标签**，**时间**等。

标题 1

标题 2

标签 1231

内容 1231

时间 2023-05-25 至 2023-06-01

+

搜索你的

实现方式具体是，在前端，会发送一个 `POST /post/searchPost` 请求，在该方法中会包装一个的 json，在SpringBoot接受到信息后，会自动转化为`List<SearchInfo>`，**SearchInfo**的具体属性为private String **select**，private String **value**，private List<Timestamp> **timeValue**；select分别有0,1,2,3等4个值用于对应参数类型的标题，内容，标签，时间。其中，只有时间使用timeValue来存储筛选的初末时间，否则值都会存在value中。在SpringBoot的Mapper层中，使用了**MyBatis**的动态sql语句，利用以下语句来遍历一种参数的值（以下为title列表的遍历）

```
1 <if test='titles != null and !titles.isEmpty()>
2     AND (
3         <foreach collection='titles' item='item' index='index' separator='
4         AND '>
5             p.title LIKE CONCAT('%', #{item}, '%')
6         </foreach>
7     </if>
```

RESTful Web 请求

PostController

该系列EndPoint都以 `/post` 开头，用以处理跟帖子有关的请求，以下只罗列 `/post` 后的部分

该系列请求中的的返回值都为**包含帖子信息的键值对列表**，类似于返回给前端`List<Post>`或者Post的效果

Post请求

路径:/searchPost

参数: `@RequestBody List<SearchInfo> searchList`，**SearchInfo**用于传递搜索需要的标题、内容、时间、目录等限制条件

作用:允许接受多个搜索限制条件，并返回相应帖子信息详细实现见“**多参数搜索部分**”

路径:/create

参数: `@RequestBody Post post`，**post**内包括了创建帖子的具体内容，有**标题**，**作者ID**，**标签**，**内容**等

作用:以**post**的内容为基础，在数据库内插入记录，创建一条帖子

路径:/sharePost

参数: @PathVariable int post_id, **post_id**为被分享的帖子的ID

作用:实现用户**分享**id为post_id的帖子的请求

路径:/userLikePost

参数: @PathVariable int post_id, **post_id**为被点赞的帖子的ID

作用:实现用户**点赞**id为post_id的帖子的请求

路径:/userDislikePost

参数: @PathVariable int post_id, **post_id**为被取消点赞的帖子的ID

作用:实现用户**取消点赞**id为post_id的帖子的请求

路径:/userFavoritePost

参数: @PathVariable int post_id, **post_id**为被收藏的帖子的ID

作用:实现用户**收藏**id为post_id的帖子的请求

路径:/userCancelFavoritePost

参数: @PathVariable int post_id, **post_id**为被取消收藏的帖子的ID

作用:实现用户**取消收藏**id为post_id的帖子的请求

路径:/uploadPic

参数: @RequestBody MultipartFile file, @PathVariable String username, **MultipartFile** 为传输的文件, **username**为传输该文件用户的名字, 用于区分不同用户上传的同名文件

作用:用于**实现视频、图片等文件传输**的请求, 会将文件存储在每个用户独立的文件夹内

Get请求

路径:/findAllPost

参数: @PathVariable int page, @PathVariable int pageSize, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数

作用:返回除被屏蔽用户外的所有帖子, 实现了分表查询, 减少了数据库和后端服务器返回数据的压力

路径:/hotList

参数: @PathVariable int page, @PathVariable int pageSize, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数

作用:返回热搜榜的帖子列表, 详细实现见“**热搜部分**”

路径:/findPostByWrite

参数: @PathVariable int page, @PathVariable int pageSize, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数

作用:返回自己**发布**的帖子列表

路径:/findPostByFavorite

参数: @PathVariable int page, @PathVariable int pageSize, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数

作用:返回自己**收藏**的帖子列表

路径:/findPostByLike

参数: @PathVariable int page, @PathVariable int pageSize, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数
作用:返回自己**点赞**的帖子列表

路径:/findPostByShare

参数: @PathVariable int page, @PathVariable int pageSize, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数
作用:返回自己**分享**的帖子列表

路径:/findByID

参数: @PathVariable Integer post_id, **post_id**表示需要查找的帖子的ID
作用:返回帖子id为id的帖子信息

路径:/getTags

参数: @PathVariable Integer post_id, **post_id**表示需要查找的帖子的ID
作用:以List<String>返回帖子的所有标签

UserController

该系列**EndPoint**都以 /user 开头

Post请求

路径:/reg

参数: @PathVariable String phone, @PathVariable String username, @PathVariable String password, **phone**表示用户注册时使用的电话号码, **username**表示用户注册时使用的用户名, **password**为用户设置的登录密码
作用:实现用户注册, 检查使用的手机号与用户名是否重复, 若均未重复则注册成功, 讲用户数据导入数据库中, 赋值一个独一无二的ID和注册时间

路径:/followUser

参数: @PathVariable String be_followed, **username**表示被关注的用户的用户名
作用:实现用户**关注其他用户的请求

路径:/cancelFollowUser

参数: @PathVariable String be_followed, **username**表示被关注的用户的用户名
作用:实现用户**取消关注其他用户的请求

路径:/blockUser

参数: @PathVariable String be_blocked, **username**表示被关注的用户的用户名
作用:实现用户**屏蔽其他用户的请求, 屏蔽时, 会取消所有对该用户的点赞、收藏操作, 并减去相应的热度

路径:/cancelBlockUser

参数: @PathVariable String be_blocked, **username**表示被关注的用户的用户名
作用:实现用户**取消屏蔽其他用户的请求

Get请求

路径:/login

参数: `@PathVariable String username, @PathVariable String password`, **username**表示用户的用户名, **password**为用户的登录密码

作用:实现用户登录, 检查用户名是否存在, 存在则核对密码是否正确, 密码正确则登录成功

路径:/findByID

参数: `@PathVariable String id`, **id**表示要查找的用户的ID

作用:返回用户ID为username的用户

路径:/findByName

参数: `@PathVariable String name`, **username**表示要查找的用户的用户名

作用:返回用户名为username的用户

路径:/findFollowList

参数: `@PathVariable int page, @PathVariable int pageSize`, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数

作用:返回当前用户**关注**的所有用户的列表

路径:/findBlockList

参数: `@PathVariable int page, @PathVariable int pageSize`, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数

作用:返回当前用户**屏蔽**的所有用户的列表

ReplyController

该系列**EndPoint**都以 `/reply` 开头, 顶级回复指直接回复该帖子本身的回复, 子回复指回复其他reply的回复

Post请求

路径:/findBlockList

参数: `@PathVariable int page, @PathVariable int pageSize`, **page**代表返回页数, **pageSize**代表返回前端展一页中的项目数

作用:返回当前用户**屏蔽**的所有用户的列表

路径:/replyToPost

参数: `@PathVariable int post_id, @PathVariable String content, @PathVariable Boolean anonymous`, **post_id**代表被回复的帖子的ID, **content**代表回复的内容, **anonymous**代表是否匿名

作用:回复**帖子**(生成顶级回复), 用户可以自选是否匿名

路径:/replyToPost

参数: `@PathVariable int parent_id, @PathVariable String content, @PathVariable Boolean anonymous`, **parent_id**代表被回复的Reply的ID, **content**代表回复的内容, **anonymous**代表是否匿名

作用:回复**Reply**(生成子回复), 用户可以自选是否匿名

Get请求

路径:/findReplyByUser

参数: @PathVariable int page, @PathVariable int pageSize, **page**代表返回页数, **pageSize**代表返回前端展示一页中的项目数

作用:返回当前用户**发布**的所有回复的列表

路径:/findTopReplyByPost

参数: @PathVariable int post_id, **post_id**代表要查找的帖子的ID

作用:返回当前帖子的**顶级回复**的列表

数据库连接池

连接池配置

我们使用的是阿里巴巴开发的**Druid**连接池, 以下是我们连接池的配置

```
spring.datasource.type=com.alibaba.druid.pool.DruidDataSource
spring.datasource.druid.max-active=20
```

连接池测试

我们利用压测软件Jmeter来进行压力测试, 每轮线程20个链接, 一共进行1500次, 每次进行2次sql请求

无连接池

Label	# 样本	平均值	最小值	最大值	标准偏差	异常 %	吞吐量	接收 KB/sec	发送 KB/sec
HTTP请求	60000	113	1	316	108.91	0.00%	164.8/sec	358.37	26.15
总体	60000	113	1	316	108.91	0.00%	164.8/sec	358.37	26.15

有连接池

Label	# 样本	平均值	最小值	最大值	标准偏差	异常 %	吞吐量	接收 KB/sec	发送 KB/sec
HTTP请求	60000	89	1	371	87.57	0.00%	221.9/sec	482.47	35.21
总体	60000	89	1	371	87.57	0.00%	221.9/sec	482.47	35.21

可以发现**有连接池**的数据库访问延迟比**无连接池**的数据库访问**快约20%**, 因此连接池的存在能够帮助数据库进行访问的优化

数据库语句优化

索引

我们在categories表中的**category**和posts中的**title**中添加了索引并进行了测试, 因为这两个属性都是我们多参数搜索中需要的用到的参数之一, 添加了索引可以帮助用户更快的进行搜索以及查询, 同时由于发帖相较于浏览的行为频次较低, 所以对title和category添加索引**相对而言**不会有较多添加的行为, 而降低索引的效率, 以下是我们创建索引的语句。

```
1 create index index_post on posts (title);
2 create index index_category on categories (category);
```

接下来我们将在post量为10000的数据库中进行测试

```

1  -- 该语句多参数搜索中常见的形式
2  explain analyse select * from posts
3      join postcategory p on posts.post_id = p.post_id
4      join category c on c.categoryid = p.categoryid
5  where c.category = '13' and posts.title like '%aa%';

```

类型	执行时间/ms	表大小/KB
有索引	0.071	1474(posts)+232(categories)
无索引	0.173	1144(posts)+168(categories)

可以看出执行时间缩短了**1/2左右**，表大小仅增加了**30%左右**，所以增加带来的空间损耗可以承受。

触发器

我们在建表时增加了三个触发器: `users_trigger`, `insert_posts_trigger`, `set_reply_id_trigger`

这三个触发器的原理与功能类似，以下以 `users_trigger` 为例

users_trigger

用户注册时，传入数据库的数据并没有用户的ID，由触发器赋值。当检测到插入的用户数据没有ID或ID为空时会为其赋值，值为当前数据库中最大的用户ID+1

```

1  IF NEW.user_id IS NULL OR TRIM(NEW.user_id) = '' THEN
2      NEW.user_id := to_char(nextval('user_id_seq'), 'FM000000000');
3  END IF;

```

触发器在插入前触发

```

1  CREATE TRIGGER users_trigger
2  BEFORE INSERT ON users FOR EACH ROW
3  EXECUTE PROCEDURE users_trigger_function();

```

前端界面

