

# CAT 二轮思路

---

## 用户

---

### 未登录

#### 查询，查看

首页展示五个问题，点赞量排序，分页

问题显示标题，提问者，和点赞量

查看问题，答案，评论。操作时弹出登录界面

可查询，模糊查询，分页

### 登录

#### 登录注册操作

##### 1. 注册与登录:

- 用户只通过邮箱方式注册账号，并向用户发送验证邮件包含验证码等信息。
  - 登录注册使用图形验证码【点击可更换验证码】
  - 忘记密码可以用邮箱发送验证码来修改密码
- 注册后有默认用户名和默认头像
- 用户可以修改个人信息
- 登录时可以使用
  - 1. 邮箱+密码
  - 2. 用户名+密码
- 忘记密码使用邮箱验证码修改密码

### 权限

权限设置:

首先，利用dao，把一个用户的所有权限都查询出来。再把查询出来的权限列表，构建成权限表，返回表

这样，用户登录后，获取权限的核心逻辑就算完成了。前端解析权限，构建页面后，用户就只能看到他有权限的东西，他没有权限的菜单或是功能，在页面上就不会展示出来。那么，为了考虑服务端的性能，用户登录一次获取了权限之后，就会把权限保存到redis，下次，用户再登录的时候，就不需要再从服务端获取权限了。但是，用户的权限，是随时会发生改变的。所以客户端的权限，要有更新的功能，问题是什么时候更新？这个时候在改变权限的时候删除redis的权限字段。

### 关注

支持关注和取消关注其他**用户、话题**等

- 可以搜索用户，并且查看用户信息，用户发布的问题贴等。
  - 关注后，可以查看与对方的共同关注列表。

## 动态界面

里面是关注的人发布的动态。

滚动分页

## 私信

里面是别人发送的私信

可以设置不接受私信，或接受已关注

## 收藏夹

里面是收藏的文章。

## 聊天室

谁创建谁群主，只有群主可以拉人。只能拉关注了群主的人

其他人可以退群

群主可以解散群

群主可以禁言群成员，群主支持添加、删除管理员。

## 自己发布问题

- 用户可以提出问题，并设置问题标题、问题内容、分类等信息，并支持图片上传等附加功能。

有人回答就有私信

## 浏览记录

记录最近五天的100条

## 拉黑

- 被答主拉黑的用户不能向答主发送**私信**，也不能在回答下面**评论**。

## 举报功能

- 可对up主的问题、问题的回答、以及回答下面的评论进行举报，需要输入举报理由。被举报的内容将会被管理员核实后不会被推送。

# 管理员

## 举报核验

- - 通过获取用户的举报信息，可以**对用户进行封禁**，被封禁的用户之后只能查看问题、回答和评论【相当于变成下边的游客身份】
  - 发布的问题需要进行审核，只有审核通过的问题才能被用户查看到（如果审核失败需要向发布者告知失败原因）
  - 管理员可以对用户进行权限管理，如发布问题、回答、点赞等权限的控制。根据用户不同的反馈进行不同的封禁，并且设定封禁截止日期，比如禁止用户评论，或者禁止用户发布问题，也有可能该用户同时不能评论或者发布问题，并且封禁的时效不一致（评论一天后解封，发布问题两天后）
  - 敏感词汇校验
    - 发布的问题、回答、以及评论，后台检测是否存在**敏感词汇**，如诋毁党、色情等信息。若存在，则系统将自动驳回发布请求并提醒发布者

# 数据库

---

## ON UPDATE CURRENT\_TIMESTAMP

MySQL自动更新列时间戳

若一个时间列设置了自动更新 `ON UPDATE`，当数据行发生更新时，数据库自动设置该列的值为当前时间

如果有更新操作、但其他列值并未变化，不会触发。此时要更新时间戳，需在SQL中指定新值给它（例如CURRENT\_TIMESTAMP）

如update table set column\_tmt = CURRENT\_TIMESTAMP

如果更新时不想触发，可显式设置时间戳列为当前值。

如update table set column\_a = 'xxx', column\_tmt = column\_tmt

## COLLATE关键字

改变

## B-TREE索引

B-TREE索引的特点

B-TREEB-TREE以B+树结构存储数据，大大加快了数据的查询速度

B-TREE索引在范围查找的SQL语句中更加适合（顺序存储）

B-TREE索引使用场景

全值匹配的查询SQL，如 where act\_id= '1111\_act'

联合索引命中匹配到最左前缀查询，如联合索引 KEY idx\_actid\_name(act\_id,act\_name) USING BTREE，只要条件中使用到了联合索引的第一列，就会用到该索引，但如果查询使用的是联合索引的第二列act\_name，该SQL则便无法使用到该联合索引（注：覆盖索引除外）

匹配模糊查询的前匹配，如where act\_name like '11\_act%'

匹配范围值的SQL查询，如where act\_date > '9865123547215'（not in和<>无法使用索引）

覆盖索引的SQL查询，就是说select出来的字段都建立了索引

**时间戳**是有13位和10位之分，其实就是毫秒和秒之分，毫秒单位的时间戳就是13位，秒单位的时间戳就是10位。

我用的是 java 开发，直接通过获取系统时间的时间戳就是13位的。**所以数据库使用bigint**

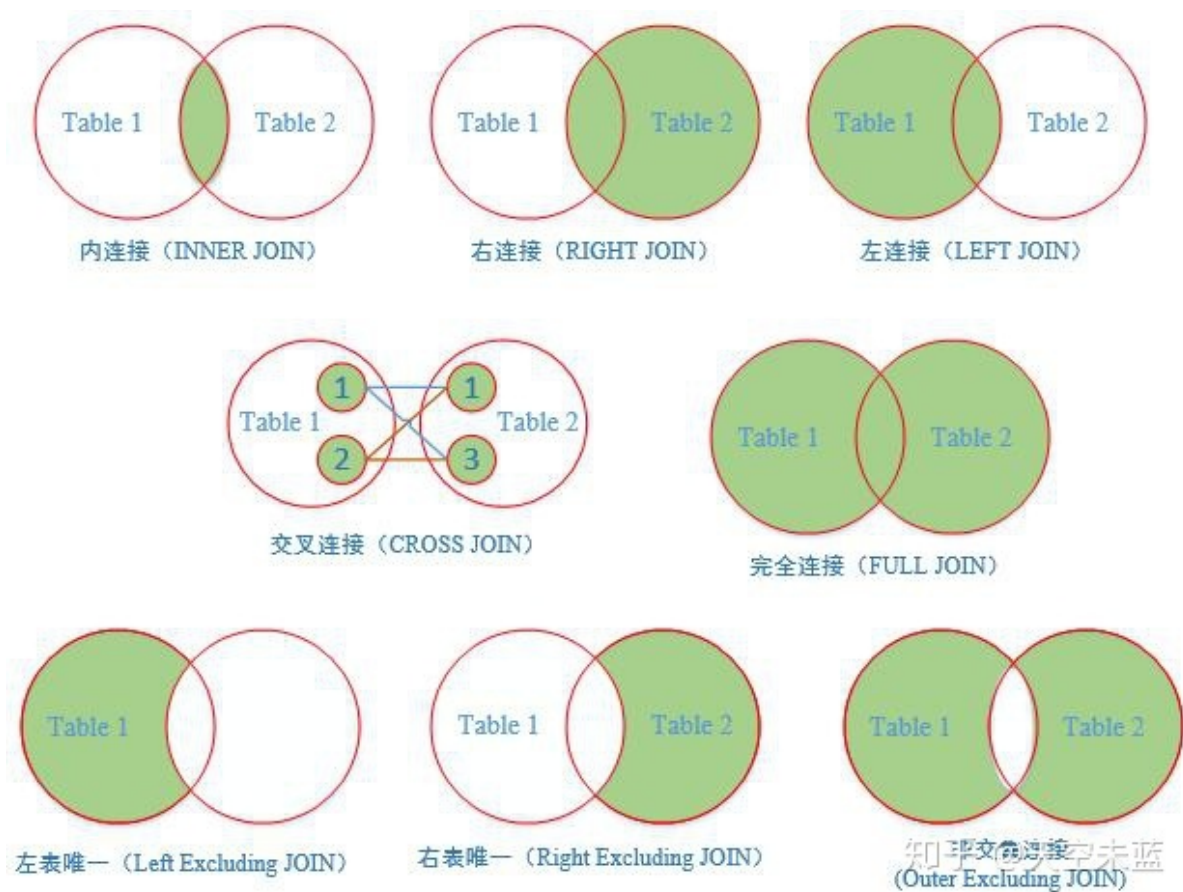
## 字段冗余

查多改少的场景，适合用字段冗余。

## 联表查询

前情提要：之前一直不敢用联表，都是单表查询+service联表，但是我看了一篇博客说，索引设好联表查询的就是O(1)

而且企业中大多数业务都是用的联表。所以现在正式开始学习联表查询



## 自连接

```
SELECT B.学号, B.姓名, B.性别, B.籍贯, B.年龄
FROM student A
JOIN student B
ON A.年龄=B.年龄 AND A.籍贯=B.籍贯 AND A.姓名='HH'
```

联表查询可以很好地解决查询多次的问题。可以减少字段冗余

内连接：

```
--查询每个学生的学号、姓名、籍贯、年龄、专业、班级
--涉及到student和major两张表，用共有字段“学号”为连接字段

--写法1: 使用INNER JOIN
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A
INNER JOIN major B
ON A.学号=B.学号

--写法2: --省去了INNER，直接写JOIN，与INNER JOIN没有区别
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A
JOIN major B
ON A.学号=B.学号

--写法3: --使用WHERE，已经逐渐被淘汰
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A, major B
WHERE A.学号=B.学号
```

--上面三种写法的结果都是一样的，推荐使用写法2

### 【LEFT JOIN】左连接:

--左连接: 显示左表student所有记录, 如右表中没有与之  
--匹配的项则以NULL值代替。

```
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A LEFT JOIN major B
ON A.学号=B.学号
```

### right join右连接:

--右连接: 显示右表major所有记录, 如左表中没有与之  
--匹配的项则以NULL值代替。

```
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A RIGHT JOIN major B
ON A.学号=B.学号
```

### 【FULL JOIN】完全连接

--完全连接: 显示两张表的并集, 如果其中一张表的记录  
--在另一张表中没有匹配的项, 则对应的数据项填充NULL

```
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A FULL JOIN major B
ON A.学号=B.学号
```

### 【Left Excluding JOIN】左表唯一

--左表唯一: 将右表B以及两张表交集的部分过滤掉,  
--得到的记录是左表中唯一存在的。

```
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A LEFT JOIN major B
ON A.学号=B.学号
WHERE B.学号 IS NULL
```

### 【Right Excluding JOIN】右表唯一

--右表唯一: 将左表A以及两张表交集的部分过滤掉,  
--得到的记录是右表中唯一存在的。

```
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A RIGHT JOIN major B
ON A.学号=B.学号
WHERE A.学号 IS NULL
```

### 【Outer Excluding JOIN】非交集连接 返回左表和右表里没有相互关联的记录。

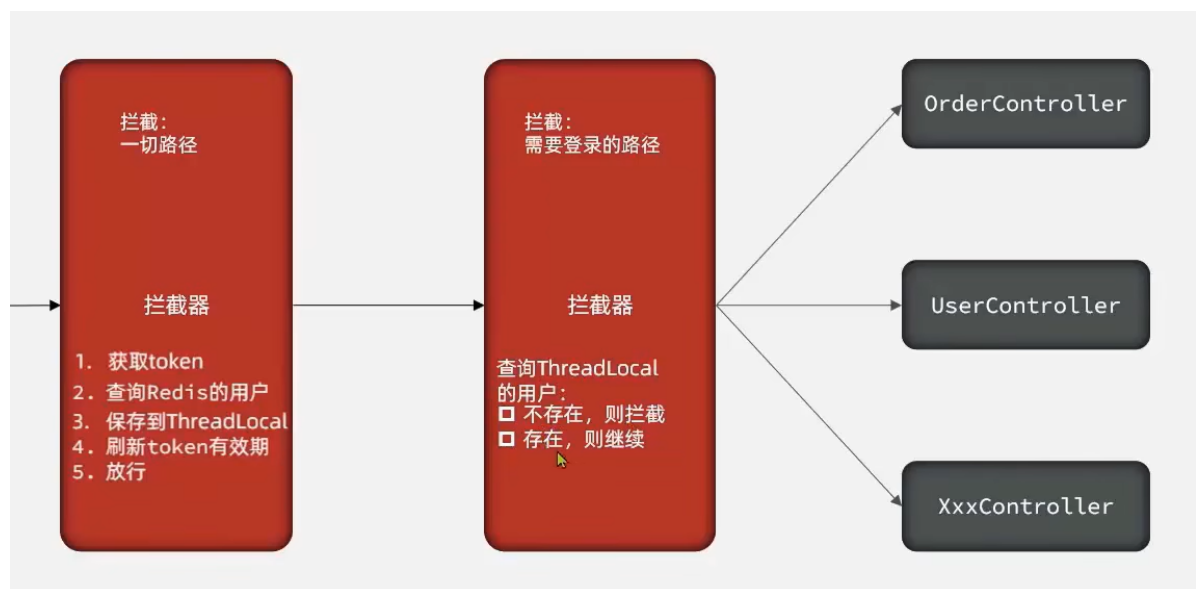
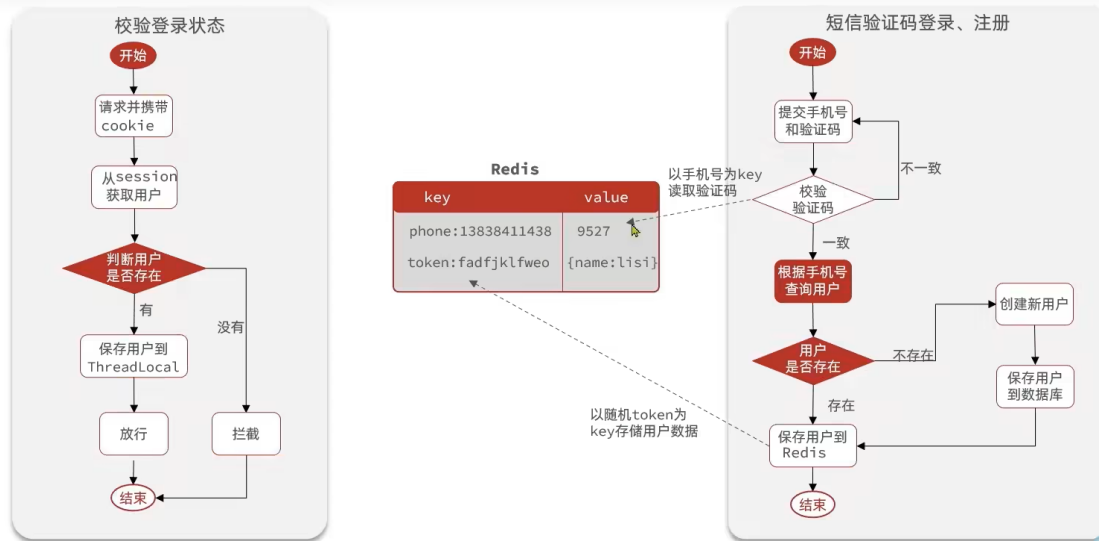
--非交集连接: 查找两张表中没有关联的记录项。

```
SELECT A.学号, A.姓名, A.籍贯, A.年龄, B.专业, B.班级
FROM student A FULL JOIN major B
ON A.学号=B.学号
WHERE A.学号 IS NULL OR B.学号 IS NULL
```

## redis

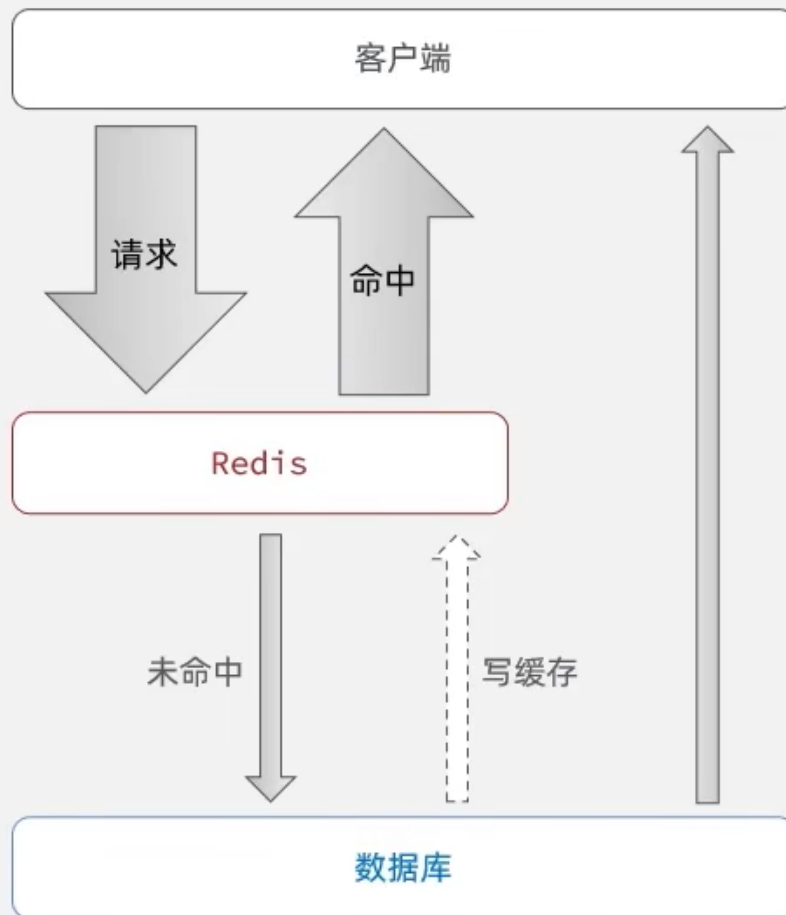
项目名:业务名:类型:id

### 基于Redis实现共享session登录

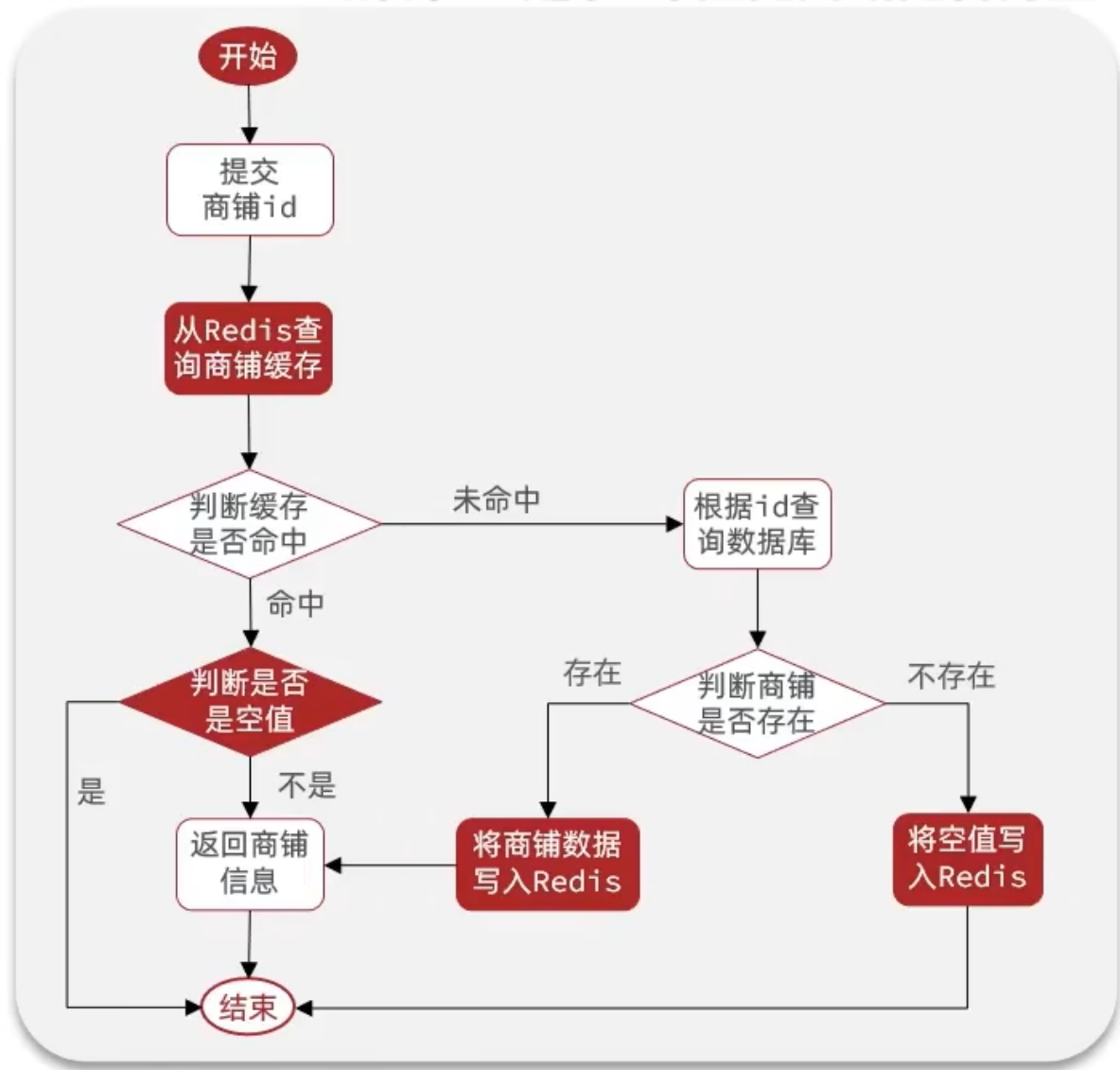


查看问题

## 缓存作用模型



缓存穿透



## 雪崩

### 缓存雪崩

缓存雪崩是指在同一时段大量的缓存key同时失效或者Redis服务宕机，导致大量请求到达数据库，带来巨大压力。

#### 解决方案：

- ◆ 给不同的Key的TTL添加随机值

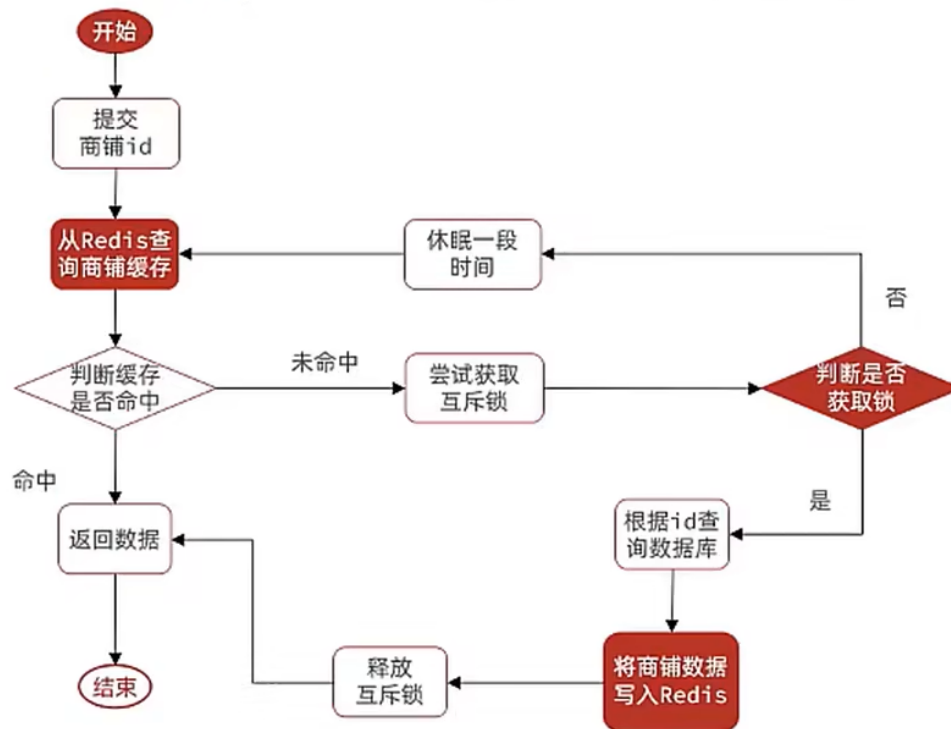
客户端

客户端

## 击穿

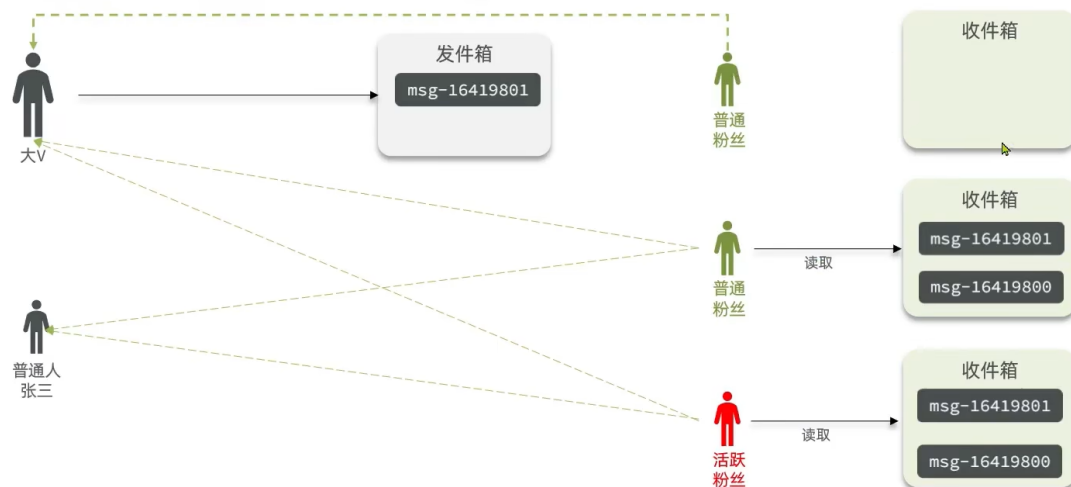


需求：修改根据id查询商铺的业务，基于互斥锁方式来解决缓存击穿问题



### Feed流的实现方案3

**推拉结合模式：** 也叫做读写混合，兼具推和拉两种模式的优点。



这里我们的项目使用推模式

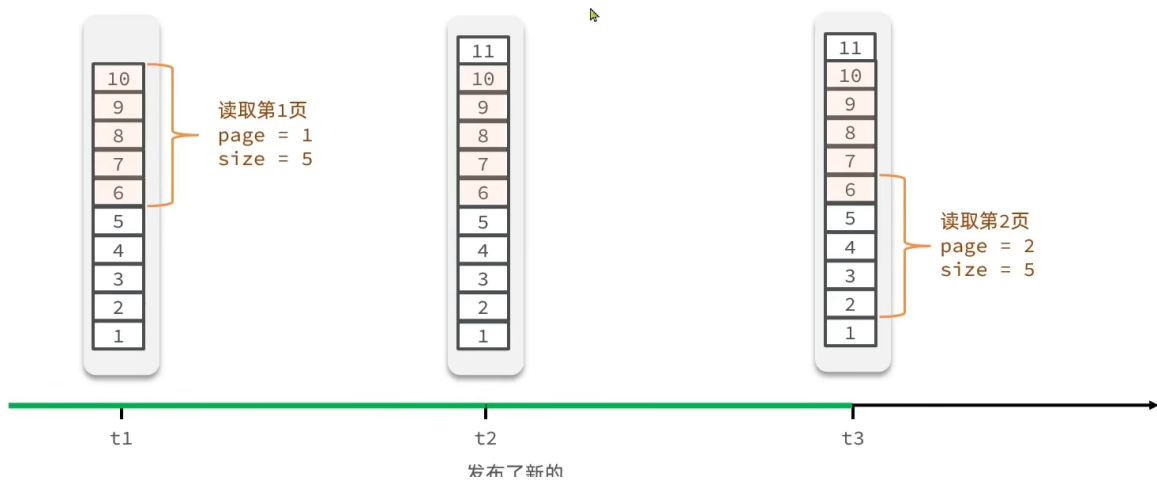
```

public Result saveBlog(Blog blog) {
    // 1. 获取登录用户
    UserDTO user = UserHolder.getUser();
    blog.setUserId(user.getId());
    // 2. 保存探店笔记
    boolean isSuccess = save(blog);
    if(!isSuccess){
        return Result.fail("新增笔记失败!");
    }
    // 3. 查询笔记作者的所有粉丝 select * from tb_follow where follow_user_id = ?
    List<Follow> follows = followService.query().eq(column: "follow_user_id", user.getId()).list();
    // 4. 推送笔记id给所有粉丝
    for (Follow follow : follows) {
        // 4.1. 获取粉丝id
        Long userId = follow.getUserId();
        // 4.2. 推送
        String key = "feed:" + userId;
        stringRedisTemplate.opsForZSet().add(key, blog.getId().toString(), System.currentTimeMillis());
    }
    // 5. 返回id
    return Result.ok(blog.getId());
}

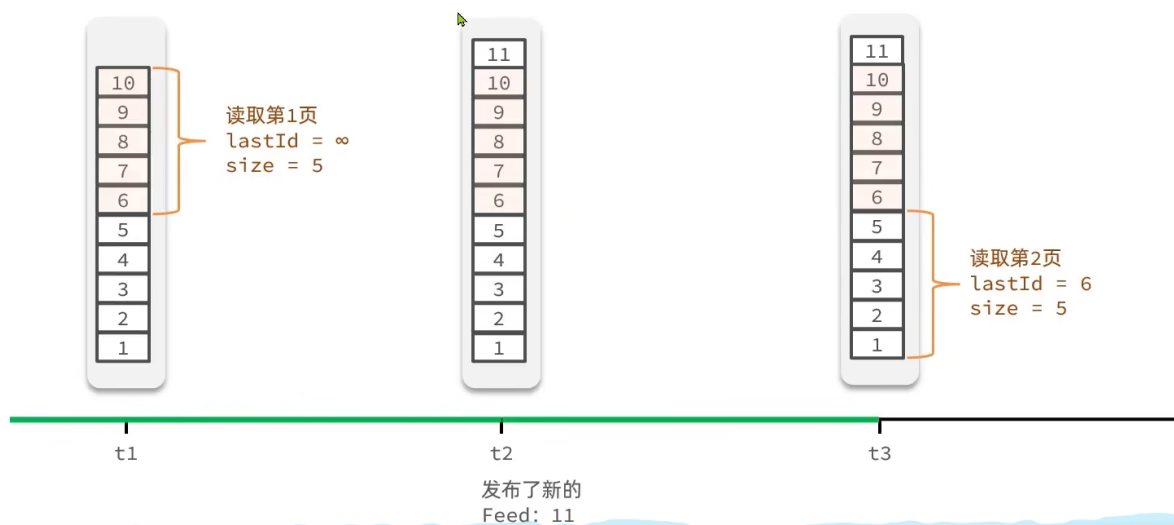
```

## Feed流的分页问题

Feed流中的数据会不断更新，所以数据的角标也在变化，因此不能采用传统的分页模式。



Feed流中的数据会不断更新，所以数据的角标也在变化，因此不能采用传统的分页模式。



使用sortedset

1 滚动分页查询参数:  
2 max: 当前时间戳 | 上一次查询的最小时间戳  
3 min: 0  
4 offset: 0 | 在上一次的结果中, 与最小值一样的元素的个数  
5 count: 3

```
}  
// 4. 解析数据: blogId、minTime (时间戳)、offset  
List<Long> ids = new ArrayList<>(typedTuples.size());  
long minTime = 0;  
int os = 1;  
for (ZSetOperations.TypedTuple<String> tuple : typedTuples) { // 5 4 4 2  
    // 4.1. 获取id  
    ids.add(Long.valueOf(tuple.getValue()));  
    // 4.2. 获取分数(时间戳)  
    long time = tuple.getScore().longValue();  
    if(time == minTime){  
        os++;  
    }else{  
        minTime = time;  
        os = 1;  
    }  
}  
// 4. 根据id查询blog
```

使用order by查询数据库, 出来有序的集合

## 日记

### 4.11

扒了点前端的代码

### 4.12

创建了数据库

### 4.13

又是没有见过的bug

```
Exception in thread "main" java.lang.NoClassDefFoundError: Create breakpoint : javax/servlet/http/HttpServlet
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(ClassLoader.java:756)
    at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142)
    at java.net.URLClassLoader.defineClass(URLClassLoader.java:473)
    at java.net.URLClassLoader.access$100(URLClassLoader.java:74)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:369)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:363) <1 个内部行>
    at java.net.URLClassLoader.findClass(URLClassLoader.java:362)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:355)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Class.java:348)
    at com.my_framework.www.utils.ClassUtil.loadClass(ClassUtil.java:38)
    at com.my_framework.www.utils.ClassUtil.doAddClass(ClassUtil.java:139)
    at com.my_framework.www.utils.ClassUtil.addClass(ClassUtil.java:115)
    at com.my_framework.www.utils.ClassUtil.addClass(ClassUtil.java:127)
    at com.my_framework.www.utils.ClassUtil.getClassSet(ClassUtil.java:63)
    at com.my_framework.www.beans.BeanDefinitionReader.<init>(BeanDefinitionReader.java:44)
    at com.my_framework.www.context.Impl.ApplicationContextImpl.refresh(ApplicationContextImpl.java:48)
    at com.my_framework.www.context.Impl.ApplicationContextImpl.<init>(ApplicationContextImpl.java:40)
```

排查了一下，发现把

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>4.0.1</version>
</dependency>
```

的删了就可以

正常运行，但是一直学的就是加上protected啊。

运行web项目时，要先包扫描，才可以注入，所以我们要在初始化tomcat的时候加上包扫描

```
'resh helloServiceImpl
'resh com.huangTaiQi.www.service.HelloService
'resh userControllerImpl
'resh com.huangTaiQi.www.controller.IUserController
'resh noArgsServiceImpl
'resh com.huangTaiQi.www.service.NoArgsService
'resh hiServiceImpl
'resh com.huangTaiQi.www.service.HiService
'ulateBean class com.huangTaiQi.www.service.impl.HelloServiceImpl autowired:com.huangTaiQi.www.service.impl.HiServiceImpl
'ulateBean 1111111null
'ulateBean class com.huangTaiQi.www.service.impl.HelloServiceImpl autowired:com.huangTaiQi.www.service.impl.HiServiceImpl
'ulateBean class com.huangTaiQi.www.controller.impl.UserControllerImpl autowired:com.huangTaiQi.www.service.impl.HelloServiceImpl
'ulateBean class com.huangTaiQi.www.controller.impl.UserControllerImpl autowired:com.huangTaiQi.www.service.impl.HelloServiceImpl
'r occurred in com.huangTaiQi.www.controller.impl.UserControllerImpl e = {0}
```

```
java.lang.reflect.InvocationTargetException <4 个内部行>
    at com.huangTaiQi.www.controller.BaseController.service(BaseController.java:39)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:582) <22 个内部行>
Caused by: java.lang.NullPointerException
    at com.huangTaiQi.www.controller.impl.UserControllerImpl.hello(UserControllerImpl.java:42)
    ... 28 more

已初始化
hello
```

为什么我已经将helloServiceImpl注入进UserController，但是还是有空指针异常？

```
field: "com.huangTaiQi.www.service.impl.HelloServiceImpl com.huangTaiQi.www.controller.impl.UserControllerImpl.helloService"
```

我明白了

```
System.out.println(this);  
System.out.println(this.equals(applicationContext.getBean( beanName: "userControllerImpl")));
```

```
com.huangTaiQi.www.controller.impl.UserControllerImpl@4b4cbd35  
false
```

不是同一个类。

我试一下能不能用在UserController用单例解决

**消息** 实例化Servlet类[com.huangTaiQi.www.controller.impl.UserControllerImpl]异常

**描述** 服务器遇到一个意外的情况，阻止它完成请求。

**例外情况**

```
javax.servlet.ServletException: 实例化Servlet类[com.huangTaiQi.www.controller.impl.UserControllerImpl]异常  
org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:494)  
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:93)  
org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:682)  
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:367)  
org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:639)  
org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:63)  
org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:932)  
org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1695)  
org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)  
org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191)  
org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659)  
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)  
java.lang.Thread.run(Thread.java:750)
```

**根本原因。**

```
java.lang.NoSuchMethodException: com.huangTaiQi.www.controller.impl.UserControllerImpl.<init>()  
java.lang.Class.getConstructor0(Class.java:3082)  
java.lang.Class.getConstructor(Class.java:1825)  
org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:494)  
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:93)  
org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.java:682)  
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:367)  
org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:639)  
org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:63)  
org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.java:932)  
org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1695)  
org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)  
org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1191)  
org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:659)  
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)  
java.lang.Thread.run(Thread.java:750)
```

我笑了。我懂了，在转发器改

## 4.14

完善beanUtils

## 4.15

如果mysql中储存了无符号的int，就会导致java中的Integer溢出，那么unsigned int被转成Long就非常合理且安全了。

## 4.18

电脑故障了，自动重启了，导致我前几天的笔记丢失了。学到了学完就保存

补一下重点的笔记

1. 解决threadlocal拿不到值的问题

因为在web项目中一个请求进来系统会在线程池中选择线程执行，所以之前在threadlocal中拿的值可能就会拿不到。

**解决办法：**在filter中将值存入threadlocal，当前请求就可以轻松地拿到了

**注意：**这就导致了一个问题，ThreadLocal在没有外部强引用时，发生GC时会被回收，如果创建ThreadLocal的线程一直持续运行，那么这个Entry对象中的value就有可能一直得不到回收，发生内存泄露。

就比如线程池里面的线程，线程都是复用的，那么之前的线程实例处理完之后，出于复用的目的线程依然存活，所以，ThreadLocal设定的value值被持有，导致内存泄露。

记得在使用的最后用remove把值清空

实现了@Transactional注解，使用cglib动态代理实现事务管理。

有一点可以优化的就是没有管方法，全部是数据库和redis都开启事务，有时候会浪费资源，

报了下面的错

```
redis.clients.jedis.exceptions.JedisException: Could not get a resource from the pool
```

这是一个很神奇的错误，因为我很清楚我的ip和密码是正确的。

所以我改了些什么会使它报错呢？看看日志好像是连接池太小了。

使用cglib动态代理实现事务管理后和jdk代理发生了相似的错误。代理对象确实不可以拿到被代理的对象的元素。

那么如何解决这个问题呢？

```
java.lang.NullPointerException Create breakpoint
  at com.huangTaiQi.www.service.impl.UserServiceImpl.login(UserServiceImpl.java:113)
  at com.huangTaiQi.www.service.impl.UserServiceImpl$$EnhancerByCGLIB$$3221b97b.CGLIB$login$1(<generated>)
  at com.huangTaiQi.www.service.impl.UserServiceImpl$$EnhancerByCGLIB$$3221b97b$$FastClassByCGLIB$$c3b79200.invoke(<generated>)
  at net.sf.cglib.proxy.MethodProxy.invokeSuper(MethodProxy.java:228)
  at com.my_framework.www.Transaction.TransactionHandler.intercept(TransactionHandler.java:22)
  at com.huangTaiQi.www.service.impl.UserServiceImpl$$EnhancerByCGLIB$$3221b97b.login(<generated>)
  at com.huangTaiQi.www.controller.impl.UserControllerImpl.login(UserControllerImpl.java:77) <4 个内部行>
  at com.huangTaiQi.www.controller.BaseController.service(BaseController.java:48)
```

cglib动态代理实现事务管理，代理类拿不到被代理类被注入的对象，因为被注入的成员变量在被代理时没有初始化，ioc是在获取到实例后再注入的，然后cglib会调用构造方法实现代理，所以cglib动态代理使用被注入的成员变量时报空指针，怎么解决？

## 4.19

乱搞git代码没了。。。。

折磨

学到了不要乱搞git，

每一个小功能都要上传

## 4.20

写请求转发

## 4.21

写请求转发

## 4.22

固定虚拟机ip, 改JK代理的bug

## 4.23

改请求转发的bug

改事务bug

如果在 Redis 的事务中使用 Jedis 命令，当调用 `multi()` 方法时，Jedis 将进入 Multi 状态。在 Multi 状态下，不能使用 Jedis 的其他方法。因此，如果尝试在这种情况下使用 Jedis 的其他方法，如 `set()`、`get()` 等，Jedis 将返回错误消息：“Cannot use Jedis when in Multi. Please use Transaction or reset jedis state.”

## 4.24

websocket

WebSocket 请求和常规的 HTTP 请求不同，不需要通过请求转发器。WebSocket 是一种基于 TCP 的协议，允许客户端和服务端之间进行双向通信。当客户端发送 WebSocket 请求时，它将直接连接到服务器，并打开一个持久化的连接，从而允许服务器向客户端推送消息，而无需客户端请求。因此，WebSocket 请求不会经过请求转发器。在使用 WebSocket 时，一般不需要在 web.xml 文件中进行特殊的配置。

## 4.25

前端限制我的发挥

## 4.26

在使用 WebSocket 实现群聊时，对于某些没有上线的用户，使用数据库来缓存他们的消息。当用户上线时，从数据库中获取这些消息并将其发送给用户。

当用户登录时，获取该用户的所在的所有群聊，推送群聊信息到 redis 中，前端通过当前所在页面的群聊拉取消息并显示。

步骤：

1. 创建一个消息队列：使用 redis 为每个用户创建一个队列，或者为每个群聊创建一个队列。
2. 当用户发送消息时，将消息添加到消息队列中。
3. 当用户上线时，从消息队列中获取消息。
4. 将消息发送给用户。

历史记录用 jedis 存七天



## 4.27

写群聊

## 4.28

写群聊

## 4.29

```
private Map<Long, List<CommentEntity>> groupCommentsByTid(List<CommentEntity>
commentList) {
    //      Map<Long, List<CommentEntity>> commentsMap = new HashMap<>();
    //
    //      // 将评论按照tid分组, 存储在commentsMap中
    //      for (CommentEntity comment : commentList) {
    //          // 如果tid在commentsMap中不存在, 则新建一个ArrayList作为value
    //          commentsMap.computeIfAbsent(comment.getTopId(), k -> new
    ArrayList<>()).add(comment);
    //      }
    //
    //      return commentsMap;
    return
commentList.stream().collect(Collectors.groupingBy(CommentEntity::getTopId));
}
```

明显看出stream的漂亮

## 4.30

写业务

## 5.1

写业务

## 5.2

写业务

## 5.3

摸鱼

## 5.4

回学校

C:\Users\14629\IdeaProjects\SECOND\_CAT\project\target\project-1.0-SNAPSHOT\img

C:\Users\14629\IdeaProjects\SECOND\_CAT\project\src\main\webapp\img

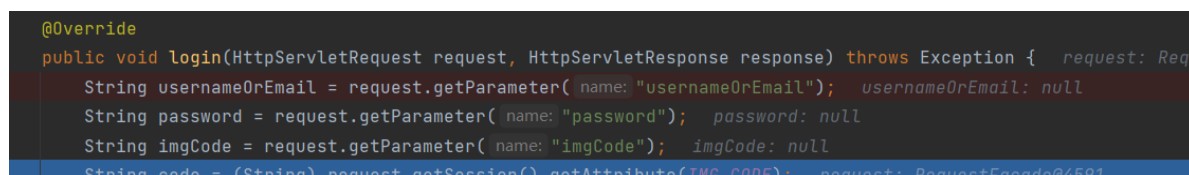
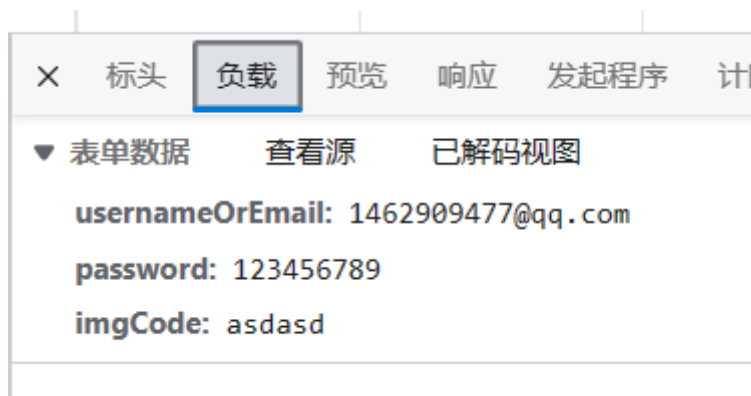
这两个路径666

`upload.parseRequest(request)` 方法是用于解析 HTTP 请求中的 `multipart/form-data` 提交的表单数据,



而在使用 `axios` 进行 POST 请求时，表单数据的格式为类似 JSON 的格式，而不是 `multipart/form-data`。

想起来了之前遇到的问题



这里可能是因为当时没有使用

`@MultipartConfig`

所以接收不到

现在真拿不到值了

`upload.parseRequest(request).size==0`

烦

本来想前端传二进制文件到后端的，但是怎么改都拿不到信息。

**imageData:** (二进制)  
**username:** 123  
**introduce:** 此用户暂时没有介  
**email:** 1462909477@qq.com  
**gender:** 0

▼ 请求标头	查看源
<b>Accept:</b> application/json, text/plain, */*	
<b>Accept-Encoding:</b> gzip, deflate, br	
<b>Accept-Language:</b> zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6	
<b>authorization:</b> token:a8d9b043-1e44-418f-a8c4-e664ab7d8cac	
<b>Cache-Control:</b> no-cache	
<b>Connection:</b> keep-alive	
<b>Content-Length:</b> 30169	
<b>Content-Type:</b> multipart/form-data; boundary=---WebKitFormBoundarypkNA9d1A9GPCiPxy	
<b>Cookie:</b> JSESSIONID=91FFD2EC101A922B8DF99EB04A13E597; Idea-56c89e28=a9d35fe6-4262-4dd0-a05e-d6470ccde79f; Idea-654d8fc6=8e3f56e4-dc24-4085-8d01-51b2e7b0520f	
<b>Host:</b> localhost:8080	

用base64试试

## 5.5

学一下ac自动机

## 5.6

写业务

## 5.7

js最无语的语言

```

.then(function (resp) {
    console.log(resp.data)
    let rightMap=JSON.parse('{ "map": '+resp.data+' }')
    console.log(rightMap)
    alert(null)
    if(resp.data["1"]===true){
        alert("json")
    }

    if(rightMap.map[1]){
        alert(1)
    }
    if(rightMap[1]){
        alert(1)
    }
    if(rightMap["1"]){
        alert(1)
    }
    if(rightMap["5"]){
        alert(1)
    }
    if(rightMap.map[5]){
        document.getElementById("ban").style.display='
    }
    if(rightMap.map[6]){

```

传个key是

数字的mapjson就懵逼了

## 5.10

写了个权限管理

### 1. 只对获取的文章做更新

在获取文章时，如果发现当前用户的冗余字段与文章表中存储的不一致，那么就可以对当前文章进行更新，将冗余字段更新为当前用户的信息。这个做法可以有效减少数据库的开销，只针对需要更新的文章进行操作，但是可能会导致文章表中冗余数据的不一致性。

### 2.对当前用户的所有文章进行更新

在获取文章时，如果发现当前用户的冗余字段与文章表中存储的不一致，那么就可以对当前用户的所有文章进行更新，将冗余字段更新为当前用户的信息。这个做法可以保证冗余数据的一致性，但是开销会比较大，需要对当前用户的所有文章进行更新。

## 5.11

```
147         } else {
148             //get请求直接获取参数
149             params = request.getParameterMap(); request: RequestFacade@4901
150         }
151
152         //TODO: 参数过滤
153         params = params.entrySet().Set<Entry<String, String[]>> params; size = 4
```

调试器 服务器 Tomcat Localhost 日志 Tomcat Catalina 日志

» 评估表达式(Enter)或添加监视(Ctrl+Shift+Enter)

han

- postData = null
- parameterMap = (ParameterMap@4920) size = 4
  - username -> (String[1]@5004) ["user\_tDxat"]
  - introduce -> (String[1]@5006) ["此用户暂时没有介绍"]
  - email -> (String[1]@5008) ["1462909477@qq.c..."]
  - gender -> (String[1]@5010) ["0"]
- parts = (ArrayList@4994) size = 5
  - 0 = (ApplicationPart@5016)
    - fileItem = (DiskFileItem@5021) \*name=IMG666.JPG, StoreLocation=C:\Users\14629\AppData\Local\JetBrains\IntelliJ\Idea2022.3\tomcat\d7fd336d-911f-42ae-a97...
    - location = (File@5022) "C:\Users\14629\AppData\Local\JetBrains\IntelliJ\Idea2022.3\tomcat\d7fd336d-911f-42ae-a973-b0328d241644\work\Catalina\localhost\pr...
  - 1 = (ApplicationPart@5017)
  - 2 = (ApplicationPart@5018)
  - 3 = (ApplicationPart@5019)
  - 4 = (ApplicationPart@5020)
- partsParseException = null

```
147         } else {
148             //get请求直接获取参数
149             params = request.getParameterMap(); request: RequestFacade@4901
150         }
151
152         //TODO: 参数过滤
153         params = params.entrySet().Set<Entry<String, String[]>> params; size = 4
```

调试器 服务器 Tomcat Localhost 日志 Tomcat Catalina 日志

» 评估表达式(Enter)或添加监视(Ctrl+Shift+Enter)

han

- postData = null
- parameterMap = (ParameterMap@4920) size = 4
  - username -> (String[1]@5004) ["user\_tDxat"]
  - introduce -> (String[1]@5006) ["此用户暂时没有介绍"]
  - email -> (String[1]@5008) ["1462909477@qq.c..."]
  - gender -> (String[1]@5010) ["0"]
- parts = (ArrayList@4994) size = 5
  - 0 = (ApplicationPart@5016)
    - fileItem = (DiskFileItem@5021) \*name=IMG666.JPG, StoreLocation=C:\Users\14629\AppData\Local\JetBrains\IntelliJ\Idea2022.3\tomcat\d7fd336d-911f-42ae-a97...
    - location = (File@5022) "C:\Users\14629\AppData\Local\JetBrains\IntelliJ\Idea2022.3\tomcat\d7fd336d-911f-42ae-a973-b0328d241644\work\Catalina\localhost\pr...
  - 1 = (ApplicationPart@5017)
  - 2 = (ApplicationPart@5018)
  - 3 = (ApplicationPart@5019)
  - 4 = (ApplicationPart@5020)
- partsParseException = null

```
DiskFileItemFactory factory = new DiskFileItemFactory(); factory: DiskFileItemFactory@4
ServletFileUpload upload = new ServletFileUpload(factory); upload: ServletFileUpload@4

// 设置最大文件大小限制 10MB
upload.setFileSizeMax(1024 * 1024 * 10);
upload.setHeaderEncoding("UTF-8");
try {
    List<FileItem> fileItems = upload.parseRequest(request); request: RequestFacade@45
    System.out.println(fileItems); fileItems: size = 0
} catch (FileUploadException e) {
    throw new RuntimeException(e);
}
}
```

```

params = request.getParameterMap(); params: size = 4
if (ServletFileUpload.isMultipartContent(request)) {
    DiskFileItemFactory factory = new DiskFileItemFactory(); factory: DiskFile
    ServletFileUpload upload = new ServletFileUpload(factory); upload: Servlet

    // 设置最大文件大小限制 10MB
    upload.setFileSizeMax(1024 * 1024 * 10);
    upload.setHeaderEncoding("UTF-8");
    try {
        List<FileItem> fileItems = upload.parseRequest(request); request: Requ
        System.out.println(fileItems); fileItems: size = 0
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

烦

redis\_报错: (error) NOAUTH Authentication required。但是我一直都是密码的。怎么突然有这个错误。

woc, 吓死我了。我刚刚因为redis的问题重启了电脑, 然后测试代码突然我的代码就一直初始化servlet。也没有报错信息什么都访问不了。我又找不到原因。以为要寄了, 搞了好久才发现数据库没开。

怎么还是有啊

```

11-May-2023 22:41:57.443 严重 [http-nio-8080-exec-8]
com.huangTaiQi.www.filter.BaseFilter.doFilter An exception occurred : NOAUTH
Authentication required.
    redis.clients.jedis.exceptions.JedisDataException: NOAUTH Authentication
required.
        at redis.clients.jedis.Protocol.processError(Protocol.java:96)
        at redis.clients.jedis.Protocol.process(Protocol.java:137)
        at redis.clients.jedis.Protocol.read(Protocol.java:192)
        at
redis.clients.jedis.Connection.readProtocolWithCheckingBroken(Connection.java:31
6)
        at redis.clients.jedis.Connection.getOne(Connection.java:298)
        at redis.clients.jedis.Connection.executeCommand(Connection.java:123)
        at redis.clients.jedis.Jedis.hgetAll(Jedis.java:5751)
        at
com.huangTaiQi.www.filter.ARefreshTokenFilter.doBeforeProcessing(ARefreshTokenFi
lter.java:37)
        at com.huangTaiQi.www.filter.BaseFilter.doFilter(BaseFilter.java:26)
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilt
erChain.java:181)
        at
org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.
java:156)
        at
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:1
67)
        at
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:9
0)
        at
org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.jav
a:494)
        at
org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:130)

```

```

        at
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:93)
        at
org.apache.catalina.valves.AbstractAccessLogValve.invoke(AbstractAccessLogValve.
java:682)
        at
org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:74)
        at
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:367)
        at
org.apache.coyote.http11.Http11Processor.service(Http11Processor.java:639)
        at
org.apache.coyote.AbstractProcessorLight.process(AbstractProcessorLight.java:63)
        at
org.apache.coyote.AbstractProtocol$ConnectionHandler.process(AbstractProtocol.ja
va:932)
        at
org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:16
95)
        at
org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:49)
        at
org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.j
ava:1191)
        at
org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.
java:659)
        at
org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:6
1)
        at java.lang.Thread.run(Thread.java:750)

```

搞不懂那里报错啊。

```
JEDIS_POOL=new JedisPool(jedisPoolConfig,"192.168.181.132",6379,1000,"redis");
```

我已经加了密码了

搞定了在每次调用jedis的时候加上auth。之前怎么没问题？

```

public static Jedis getResource(){
    Jedis jedis = JEDIS_POOL.getResource();
    jedis.auth("redis");
    return jedis;
}

```

拿jedis时再写一次密码

com.alibaba.fastjson.JSON的toJSONString为空问题，没有get方法。之前都写了get，所以不知道。

## 5.12

再说吧。用parts = request.getParts();搞定了文件上传

## 5.13

写业务

## 5.14

```
com.my_framework.www.exception.ParameterInvalidException Create breakpoint : 参数不符合要求, 请检查后重试
    at com.my_framework.www.webmvc.HandlerAdapter.handle(HandlerAdapter.java:182)
    at com.my_framework.www.webmvc.DispatcherServlet.doDispatch(DispatcherServlet.java:155)
    at com.my_framework.www.webmvc.DispatcherServlet.doPost(DispatcherServlet.java:54)
    at com.my_framework.www.webmvc.DispatcherServlet.doGet(DispatcherServlet.java:47)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:488)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:582) <5 个内部行>
    at com.huangTaiQi.www.filter.BaseFilter.doFilter(BaseFilter.java:29) <19 个内部行>
com.my_framework.www.exception.ParameterInvalidException Create breakpoint : 参数不符合要求, 请检查后重试
    at com.my_framework.www.webmvc.HandlerAdapter.handle(HandlerAdapter.java:182)
    at com.my_framework.www.webmvc.DispatcherServlet.doDispatch(DispatcherServlet.java:155)
    at com.my_framework.www.webmvc.DispatcherServlet.doPost(DispatcherServlet.java:54)
    at com.my_framework.www.webmvc.DispatcherServlet.doGet(DispatcherServlet.java:47)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:488)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:582) <5 个内部行>
    at com.huangTaiQi.www.filter.BaseFilter.doFilter(BaseFilter.java:29) <19 个内部行>
```

参数验证