springboot整合shiro-SimpleAuthenticationInfo 参数该使用username 还是User实体(十八)

前言

相信大家在学习 Shiro 的时候,肯定免不了在网上百度搜索资料,在看了很多篇博客之后,肯定也发现一个问题，在 ShiroRealm 的 doGetAuthenticationInfo 方法返回值 SimpleAuthenticationInfo 中,第一个参数 有的传的是 username 而有的传的是 User实体 ,大家会奇怪,但是为了尽快完成需求，可能会暂时忽略这个问题,最后也不了了之啦，反正能用就行。

介绍一些其他的东西

首先介绍一下 redis 中 两个 key 有的人对 redis 缓存的几个 key 是干什么的还不是很清楚

shiro:cache:authenticationCache:test 和 shiro:cache:authorizationCache:test 这两个缓存。

shiro :cache:authenticationCache:test

它是身份认证的缓存, ShiroRealm 中的 doGetAuthenticationInfo 执行之后缓存的结果。

如果是正常的调用登出操作,这个 缓存会自动清除，如果非正常情况(刚登陆之后,手动清理浏览器缓存)，这个 key 依然保留在 redis 中。

如果 redis 中有这个缓存,当你下次登录的时候 ， 将不再执行 doGetAuthenticationInfo 而是从缓存中获取该认证,然后比较密码。

这就会出现一个问题,假如说 用户修改了密码,但是 你却没有清除 这个缓存,那么再次登录的时候,新密码就不能用，还需要使用之前的老密码登录,因为 redis 中缓存的是之前老密码的认证。

所以,在执行修改密码，或者 支付密码之类的操作，要清理掉 该缓存 或者 干脆就不缓存。

shiro:cache:authorizationCache:test

它是权限认证的缓存, ShiroRealm 中的 doGetAuthorizationInfo 执行之后缓存的结果。它是将数据库中角色 和 权限 查出来,然后分别放到一个 Set 里，然后序列化 到 redis 中。

当你访问一个 url 的时候,会调用 ShiroRealm 的 isPermitted(PrincipalCollection principals, String permission); 方法判断用户是否有这个权限。如果没有就会抛出异常。

例如 使用 test 登录，访问 http://localhost:9090/userInfo/clearCache test 的用户没有清理其他用户缓存的权限，访问控制台打印:

```
org.apache.shiro.authz.UnauthorizedException:
Subject does not have permission [userInfo:clearCache]
```

如果 redis 中有这个缓存,那就不会查询数据库,直接从缓存中获取进行判断。

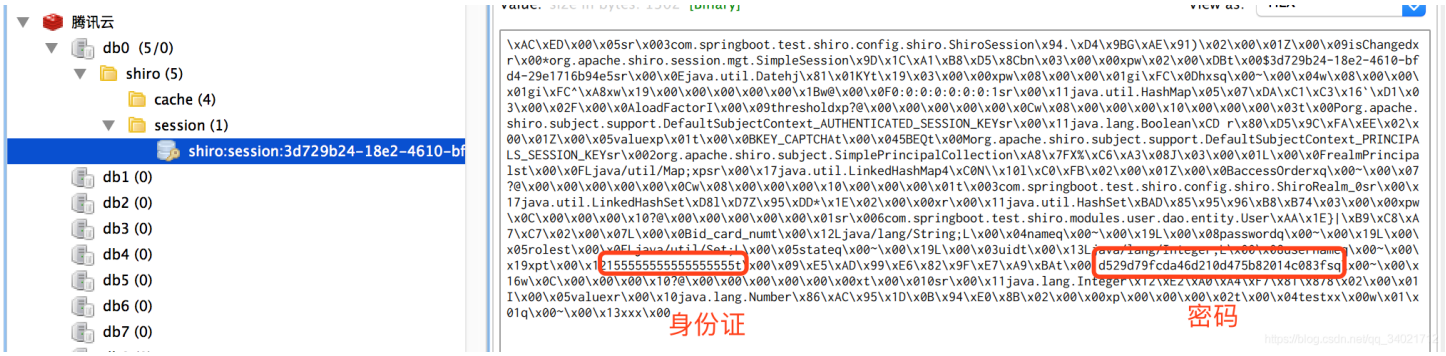
所以在给某个用户添加新的权限之后,要删除这个缓存,否则新分配的权限无法生效。

接着介绍本文的主题

看完上面的介绍,我相信已经很清楚这两个 key 是干啥的吧，以后就不要再问我了， 下面回到我们这篇文章的主题, SimpleAuthenticationInfo 的第一个参数到底是该使用 username 还是 User 实体。下面说一个 网友遇到的小问题。

这名网友呢,他在 SimpleAuthenticationInfo 中传的是 User 实体， 他认为 shiro:cache:authenticationCache:test 缓存的就是 User 的信息， 然后使用 User user = (User)SecurityUtils.getSubject().getPrincipal(); 来获取用户的信息，他跟我说,删除 shiro:cache:authenticationCache:test 这个 key 之后，依然可以获取到 User 的信息.也没有查询数据库啊,怎么拿到的信息?

其实呢,这个 User 在 authenticationCache 中保存了一份,在 session 中保存了一份, 你可以看一下你的 session 信息,会发现里面有 User 的信息。如下图:



当你使用 SecurityUtils.getSubject().getPrincipal(); 是从 session 中获取的信息，authenticationCache 那一份其实并没有什么用，所以清理掉 authenticationCache 依然可以获取到 用户的信息。

而当用户信息进行修改后，如果你再使用 `SecurityUtils.getSubject().getPrincipal();` 来获取缓存中的用户信息， 就会产生脏读,数据是之前的老数据,而如果要删除的话, 你又不知道 该用户的 `sessionId` 是多少,所以无法清理这个缓存。

是username 还是User实体?

我们先来看一下 `SimpleAuthenticationInfo` 的第一个参数是 `principal` ， `principal` 是什么呢？如果阅读源码你会看到如下的注释：

```
public interface PrincipalCollection extends Iterable, Serializable {

    /**
     * Returns the primary principal used application-wide to uniquely identify the owning account/Subject.
     * <p/>
     * The value is usually always a uniquely identifying attribute specific to the data source that retrieved the
     * account data.  Some examples:
     * <ul>
     * <li>a {@link java.util.UUID UUID}</li>
     * <li>a {@code long} value such as a surrogate primary key in a relational database</li>
     * <li>an LDAP UUID or static DN</li>
     * <li>a String username unique across all user accounts</li>
     * </ul>
     * <h3>Multi-Realm Applications</h3>
     * In a single-{@code Realm} application, typically there is only ever one unique principal to retain and that
     * is the value returned from this method.  However, in a multi-{@code Realm} application, where the
     * {@code PrincipalCollection} might retain principals across more than one realm, the value returned from this
     * method should be the single principal that uniquely identifies the subject for the entire application.
```


`principal` 参数可以是uuid， 数据库主键，LDAP UUID或静态DN 或者是用户唯一的用户名。 所以说这个值 必须唯一，你可以选择 邮箱，或者 手机号，身份证号等等。

根据上面解释的情况,所以,我建议使用 `username` ， 而不是选择 `User` 实体，对于用户的 信息缓存,单独在redis中进行缓存,不要跟shiro的掺和到一起。

关于将项目中的 `User` 改为 `username` 之后 报异常无法启动项目,可以参考我上篇博客获得源码 参考更改,
博客地址：springboot整合shiro-关于登出时,redis中缓存没有清理干净的问题(十七)