java

搜索

会员中心 🎁 消息

(专栏)

Spring Security: 授权GrantedAuthority介绍



在之前的博客中,已经介绍了 Spring Security 的用户 UserDetails 、用户服务 UserDetailsService 和密码编码器 PasswordEncoder ,它们都是用于验份,而 GrantedAuthority 则表示用户验证通过后被授予的权限(可能授予多种权限),本篇博客介绍 GrantedAuthority 接口及其实现类。

- Spring Security: 用户UserDetails源码与Debug分析
- Spring Security: 用户服务UserDetailsService源码分析
- Spring Security: 密码编码器PasswordEncoder介绍与Debug分析

应用的依赖:

启动类:

```
1  @SpringBootApplication
2  public class Application {
3    public static void main(String[] args) {
4         SpringApplication.run(Application.class);
5    }
6 }
```

GrantedAuthority

GrantedAuthority 接口源码:

```
package org.springframework.security.core;
3
   import java.io.Serializable;
4
   import org.springframework.security.access.AccessDecisionManager;
6
7
    * 表示授予Authentication对象 (需要进行验证或通过验证的用户封装) 的权限
8
   public interface GrantedAuthority extends Serializable {
10
11
12
        * 获取权限
13
14
15
       String getAuthority();
                                         ITKaven ( 关注 )
                                                                                                                     专栏
16
```

GrantedAuthority 接口及其实现类,如下图所示:



SimpleGrantedAuthority

GrantedAuthority 的基本具体实现,存储授予 Authentication 对象的权限的 String 表示形式。

SimpleGrantedAuthority 类源码:

```
1 | public final class SimpleGrantedAuthority implements GrantedAuthority {
 2
 3
        private static final long serialVersionUID = SpringSecurityCoreVersion.SERIAL_VERSION_UID;
 4
 5
        // 授予Authentication对象的权限
        private final String role;
 6
 8
        public SimpleGrantedAuthority(String role) {
 9
            Assert.hasText(role, "A granted authority textual representation is required");
10
            this.role = role;
11
12
13
        @Override
14
        public String getAuthority() {
            return role;
15
16
17
        // 只会比较role属性是否equals
18
19
20
        public boolean equals(Object obj) {
21
            if (this == obj) {
22
                return true;
23
24
25
            if (obj instanceof SimpleGrantedAuthority) {
                return role.equals(((SimpleGrantedAuthority) obj).role);
27
            }
28
29
            return false;
30
        }
31
32
        @Override
33
        public int hashCode() {
34
            return this.role.hashCode();
35
36
37
        @Override
        public String toString() {
38
39
            return this.role;
40
41 }
```

增加配置文件:

```
1 spring:
2 security:
3 user:
4 name: kaven
5 password: itkaven
6 roles:
7 - USER
8 - ADMIN
```









Debug 启动应用,Spring Security 在应用启动时会创建配置文件中定义的用户,首先会创建用户服务(InMemoryUserDetailsManager) bean 。

```
| Second | Conditional Conditi
```

通过用户服务创建用户,并且授予权限(通过创建 SimpleGrantedAuthority 实例)。

```
public DerBuilder roles(String...roles) { roles [rusex*, *AURIN*] | roles (string role : roles) { roles (rusex*, *AURIN*) | role : rusex* | roles (rusex*, *AURIN*) | roles (rusex*, *AUR
```

可见,SimpleGrantedAuthority是默认的授权实现(特殊场景除外),它只存储权限,是一种简易的授权实现。

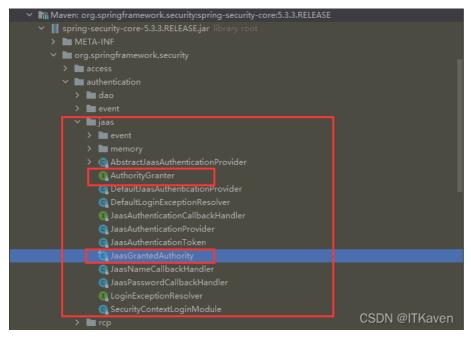
JaasGrantedAuthority

JaasGrantedAuthority 类源码:

```
1
2
    * 除了分配的角色,还持有授权人(AuthorityGranter)的主体(Principal),用作授予此权限的理由
3
4
   public final class JaasGrantedAuthority implements GrantedAuthority {
5
6
       private static final long serialVersionUID = SpringSecurityCoreVersion.SERIAL_VERSION_UID;
7
8
       private final String role;
       // 授权人的主体
9
       private final Principal principal;
10
```

AuthorityGranter 接口源码:

```
4
   public interface AuthorityGranter {
5
6
       * 根据主体映射到角色名称集合
7
8
9
      Set<String> grant(Principal principal);
```



JaasGrantedAuthority 是一种用于 Java 验证和授权服务(Java Authentication and Authorization Service,简称 JAAS)场景下的授权实现,感兴 解 JAAS 。

SwitchUserGrantedAuthority

SwitchUserGrantedAuthority 类源码:

```
1 /**
    * SwitchUserFilter使用的自定义GrantedAuthority
    * 存储原始用户的Authentication对象,以便从退出用户切换时使用。
5
    public \ final \ class \ SwitchUserGrantedAuthority \ implements \ GrantedAuthority \ \{ \\
6
7
       private static final long serialVersionUID = SpringSecurityCoreVersion.SERIAL_VERSION_UID;
8
9
       private final String role;
       // 存储原始用户的Authentication对象
10
```

该授权实现类似于 Linux 系统下用户之间的切换授权(使用 su 命令的权限),如下所示在 Linux 系统中添加 kaven 用户,然后从 root 用户切换到 kav (root 用户有这个权限)。

```
1 root@48556522ad65:~# adduser kaven
2
    Adding user `kaven' ...
    Adding new group `kaven' (1000) ...
    Adding new user `kaven' (1000) with group `kaven' ...
    Creating home directory `/home/kaven' ...
    Copying files from `/etc/skel' ...
6
    New password:
8
    Retype new password:
    passwd: password updated successfully
9
   Changing the user information for kaven
10
11 | Enter the new value or mose FMTER for the default
```









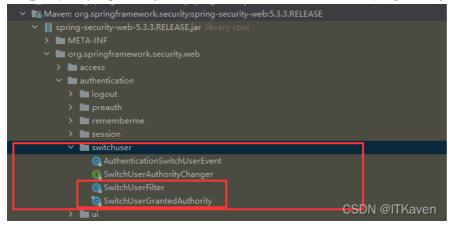






 \sim

SwitchUserGrantedAuthority 是 SwitchUserFilter 使用的自定义 GrantedAuthority , SwitchUserFilter (用户切换处理过滤器)负责用户上下文切损 Spring Security 管理的 web 应用程序,此过滤器类似于 su 命令,此功能的一个常见例子是能够允许更高权限的用户(如 ROLE_ADMIN)切换到普通FROLE_USER)。 Spring Security 的过滤器以及如何将自定义的过滤器集成到 Spring Security 中,博主以后会进行介绍,这里只是了解即可。



所以,SwitchUserGrantedAuthority 是一种用于用户切换场景下的授权实现,不仅存储了权限,还存储了原始用户的 Authentication 对象,即 source 便用户切换的退出。

```
1 kaven@48556522ad65:/root$ exit
2 exit
3 root@48556522ad65:~# exit
4 logout
5 Connection closing...Socket close.
6
7 Connection closed by foreign host.
8
9 Disconnected from remote host(predict) at 13:40:23.
10
11 Type `help' to learn how to use Xshell prompt.
12 [C:\~]$
```

Spring Security 的授权 GrantedAuthority 介绍就到这里,如果博主有说错的地方或者大家有不同的见解,欢迎大家评论补充。

文章知识点与官方知识档案匹配,可进一步学习相关知识

Java技能树 首页 概览 149725 人正在系统学习中

AI无限•码力全开, 立即参与通义灵码开源大赛, 赢灵码限量周边!

>>立即参与

1条评论



AdingLau 热评 收下我的膝盖 Orz!

Spring Security(五)--管理用户_spring security userdetails

package com. mbw.pojo; importorg. spring framework. security. core. Granted Authority; importorg. spring framework. security. core. user details. User Details; import java. util. Array Label to the properties of the properties

SpringSecurity-6-GrantedAuthority/AccessDeniedHandler接口(权限菜...

{TestUser user=newTestUser();user.setUsername("用户2");user.setPassword(encoder.encode("123456"));user.setAge(19);user.setBirthday("2005-09-15");// 设置权限,user.setPassword(encoder.encode("123456"));user.setAge(19);user.setBirthday("2005-09-15");// 设置权限,user.setAge(19);user.setBirthday("2005-09-15");// 设置权限,user.setAge(19);user.setAge(19);user.setBirthday("2005-09-15");// 设置权限,user.setAge(19);user.setAge(19);user.setAge(19);user.setAge(19);user.setBirthday("2005-09-15");// 设置权限,user.setAge(19);user.setAge(19)

【详解】Spring Security的GrantedAuthority (已授予的权限)

蔡小波的

转自: https://www.cnblogs.com/longfurcat/p/9417422.html 感谢大佬 前言

这篇是很久之前学习Spring Security整理的博客,发现浏览量都1000多了,一个赞都没有,

Spring Security API: GrantedAuthority

dengdeying的

GrantedAuthority Declared package org.springframework.security.core; public interface GrantedAuthority extends Serializable Class Jdoc 表示授权给用户的权限。 Gran

Spring Security 身份验证的基本类/架构 security contextholder-CSDN...

StringgetAuthority();//返回的是一个字符串Stringrole } //所以, GrantedAuthority, 这个玩意是用来授权的, 存储已认证用户具备的访问权限。 通过Authentication. getAuthorities

Spring Security(15)——权限鉴定结构_grantedauthority

Spring Security内置了一个GrantedAuthority的实现,Sim













Spring Security自定义GrantedAuthority前缀

Spring Security: 授权GrantedAuthority介绍-CSDN博客

如果我们正使用Spring Security提供默认的基于方法的权限认证注解如下: @PreAuthorize("hasAnyRole('ADMIN', 'USER')") public void moveTo(String id, String parentlc

Spring Security (15) ——权限鉴定结构

dotedy的t

Spring Security角色继承AffirmativeBasedVoter权限鉴定 权限鉴定结构 目录 1.1 权限 1.2 调用前的处理 1.2.1 AccessDecisionManager 1.2.2 基于投票的Acces

Spring Security-GrantedAuthority已授予的权限、权限检查 spring se...

在Security中,角色和权限共用GrantedAuthority接口,唯一的不同角色就是多了个前缀"ROLE_",而且它没有Shiro的那种从属关系,即一个角色包含哪些权限等等。在Security

那些年SpringSecurity踩过的坑 GrantedAuthority could not read js...

那些年SpringSecurity踩过的坑 GrantedAuthority 前言 在我学习springsecurity的过程中使用它进行用户授权时出现了几个错误,这几个错误有json异常Could not read JSON

Spring Security OAuth2 自定义GrantedAuthority授权接口

OceanSkv的

使用security oatuh2的时候需要返回给前端用户的角色或者权限,框架提供了GrantedAuthority接口,有一个默认的实现SimpleGrantedAuthority,但是它只能返回简单的:

SpringSecurity中的UserDetails 和 GrantedAuthority

qq 43011496的

1. UserDetails UserDetails 是一个接口,同时也是要开发者去自定义实现的。开发者要根据自己的需求来实现对应的方法。 GrantedAuthority

...一】核心组件_springsecurity的grantedauthority的作用

GrantedAuthority, to reflect the application-wide permissions granted to a principal,反映授予主体的应用程序范围的权限。 UserDetails, to provide the necessary informati

SpringSecurity学习(七)授权

Huathy的

SpringSecurity:认证与<mark>授权</mark>,基于URL的权限管理、基于方法的权限管理。权限表达式、<mark>授权</mark>原理分析。<mark>授权</mark>实战——权限模型

SpringSecurity原理(三)——授权

trayvontang的

概述 前面,我们已经简单的<mark>介绍</mark>了一下校验用户名和密码的认证过程。 这里,我们来了解一下Spring Security的权限校验过程。 首先,我们还是先通过一个简单的示例,

【详解】GrantedAuthority(已授予的权限)

diegiuxie4160的

前言 这篇是很久之前学习Spring Security整理的博客,发现浏览量都1000多了,一个赞都没有,那说明写得确实不怎么样,哈哈。应该很多初学者对这个接口存在疑

springsecurity权限验证的时候 出现了 SimpleGrantedAuthority无法序列化报空指针的问题

weixin 45699541的

在做springsecurity权限验证的时候出现了SimpleGrantedAuthority无法序列化报空指针的问题。但从redis中取出authorityList的时候无法取出里面的数据。我的工具是Strin

【项目实训】Spring Security授权

par ser的

<mark>授权</mark>是web应用常见的需求,我们需要针对不同的用户,根据他们不同的权限,放行相应的接口。在SpringSecurity中,会使用默认的FilterSecurityInterceptor来进行权限标识。

spring boot 集成 security 授权

avavazzz的

授权上一篇写了认证,授权其实就是把数据库查出来的角色、权限交给security来管理对比 1.准备工作 用户、用户-角色、角色、角色-权限、权限 2.授权 首先从数据库中i

Spring Security简单的登陆验证授权

shanving0324的

Spring Security的介绍就省略了,直接记录一下登陆验证授权的过程。 Spring Security的几个重要词 1.SecurityContextHolder: 是安全上下文容器,可以在此得知操作的F

Spring Boot Security配置用户认证和资源授权(URL与角色)

swg321321的

Spring Boot Security:完成用户认证,资源授权配置(一般是用户角色和URL的匹配)。

如何利用SpringSecurity进行认证与授权 最新发布

qq 63218110的t

本篇博客主要<mark>介绍SpringSecurity</mark>框架的学习,主要包含快速入门,以及认证、<mark>授权</mark>等方面的<mark>介绍</mark>,还涉及一些简单的自定义<mark>授权</mark>校验方法。

springboot3整合SpringSecurity实现登录校验与权限认证(万字超详细讲解) 热门推荐

2301 78646673的

用户提交登录请求Spring Security 将请求交给 UsernamePasswordAuthenticationFilter 过滤器处理。UsernamePasswordAuthenticationFilter 获取请求中的用户名和密码,

SpringSecurity授权

凉茶铺的

在第二部分中我们讲解的都是用户认证,不管是用户名密码,还是图形验证码等,最终的目的都是一个让系统知道你到底是谁在访问你的系统,解决的问题是,你是谁?这部分主具

关于我们 招贤纳士 商务合作 寻求报道 ☎ 400-660-0108 図 kefu@csdn.net ⑤ 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照 ©1999-2024北京创新乐知网络技术有限公司



ITKaven 码龄7年

♥ 暂无认证

542 2万+ 105万+ 227万+ 原创 周排名 总排名 等级 访问

2万+ 1368 1992 937 3844 积分 粉丝 获幣 评论 收藏































私信

关注



搜博主文章

Q

热门文章

MyBatis-Plus 之分页查询 ① 167673

同步和异步的区别 ① 89904

怎么保存退出 vim 编辑 ① 70673

Spring 中 ClassPathXmlApplicationContext

类的简单使用 ◎ 40177

分类专栏



最新评论

ElasticJob-Lite: 作业监听器

一只烤鸭朝北走啊: 好久没看了, 有点忘记 了, 我那会儿做的是定时任务在分布式 ...

ElasticJob-Lite: 作业监听器

AKKO111: 那感觉没必要用这个Distribute的 listener了,直接用ElasticJobServiceLoa

Nginx: Nginx添加SSL实现HTTPS访问 风度翩翩609: 我这边配完 http+域名可以,

但是https+域名就不行了

使用Vue和Spring Boot实现文件下载 一入程序无退路: 下载按钮点第二次就不太 好使怎么回事

Docker容器中使用PING命令报错: bash:... 桂哥317: 干脆利落、实测有效

最新文章

MySQL: 备份 & 导入备份















RabbitMQ: Docker Compose部署RabbitMQ

集群

VirtualBox: 系统安装重启后又要重新安装

 2022年
 80篇
 2021年
 55篇

 2020年
 206篇
 2019年
 10篇

 2018年
 220篇
 2017年
 31篇



目录 GrantedAuthority SimpleGrantedAuthority JaasGrantedAuthority SwitchUserGrantedAuthority











