



ITKaven

关注



11



27



1





专栏

Spring Security: 授权GrantedAuthority介绍

原创ITKaven 于 2022-01-06 14:08:48 发布 阅读量7k 收藏 27 点赞数 11

分类专栏: Spring Security 文章标签: spring java 后端

 华为云开发者空间 该内容已被华为云开发者联盟社区收录

 Spring Security 专栏收录该内容

31 订阅 12 篇文章

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**，它们都是用于验证，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:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5     <modelVersion>4.0.0</modelVersion>
6
7     <groupId>com.kaven</groupId>
8     <artifactId>security</artifactId>
9     <version>1.0-SNAPSHOT</version>
10
11     <parent>
```

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:

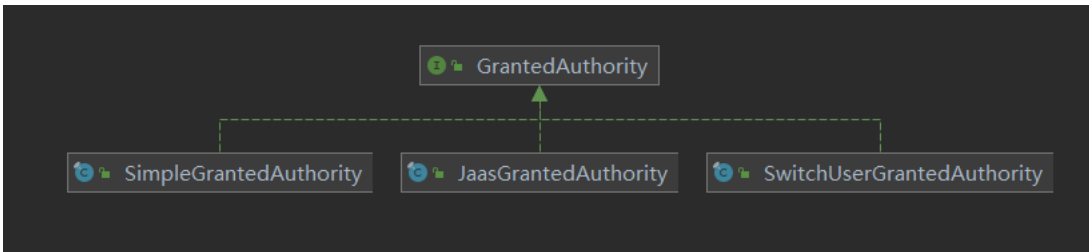
```
1 package org.springframework.security.core;
2
3 import java.io.Serializable;
4
5 import org.springframework.security.access.AccessDecisionManager;
6
7 /**
8  * 表示授予Authentication对象（需要进行验证或通过验证的用户封装）的权限
9  */
10 public interface GrantedAuthority extends Serializable {
11
12     /**
13      * 获取权限
14      */
15     String getAuthority();
16 }
```

 **ITKaven** 关注

11 27 1

专栏

`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
6     private final String role;
7
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() {
15        return role;
16    }
17
18    // 只会比较role属性是否equals
19    @Override
20    public boolean equals(Object obj) {
21        if (this == obj) {
22            return true;
23        }
24
25        if (obj instanceof SimpleGrantedAuthority) {
26            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
38    public String toString() {
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
```



ITKaven

关注

11



27

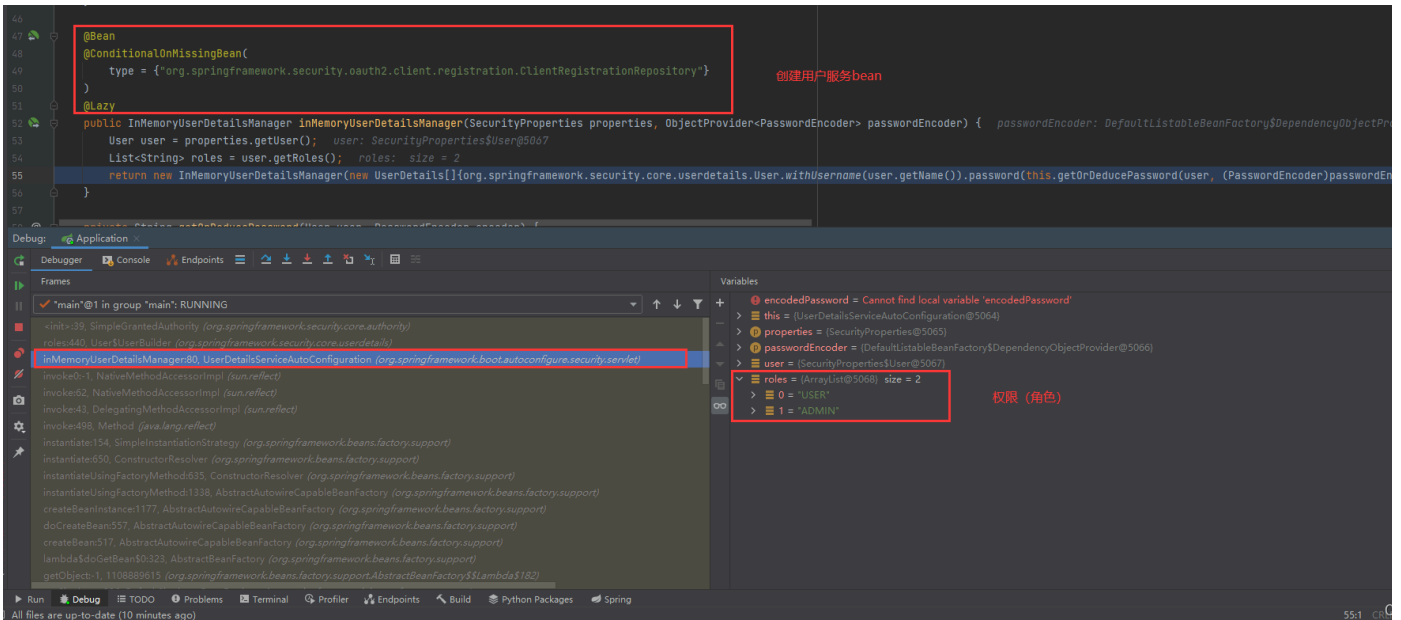


1

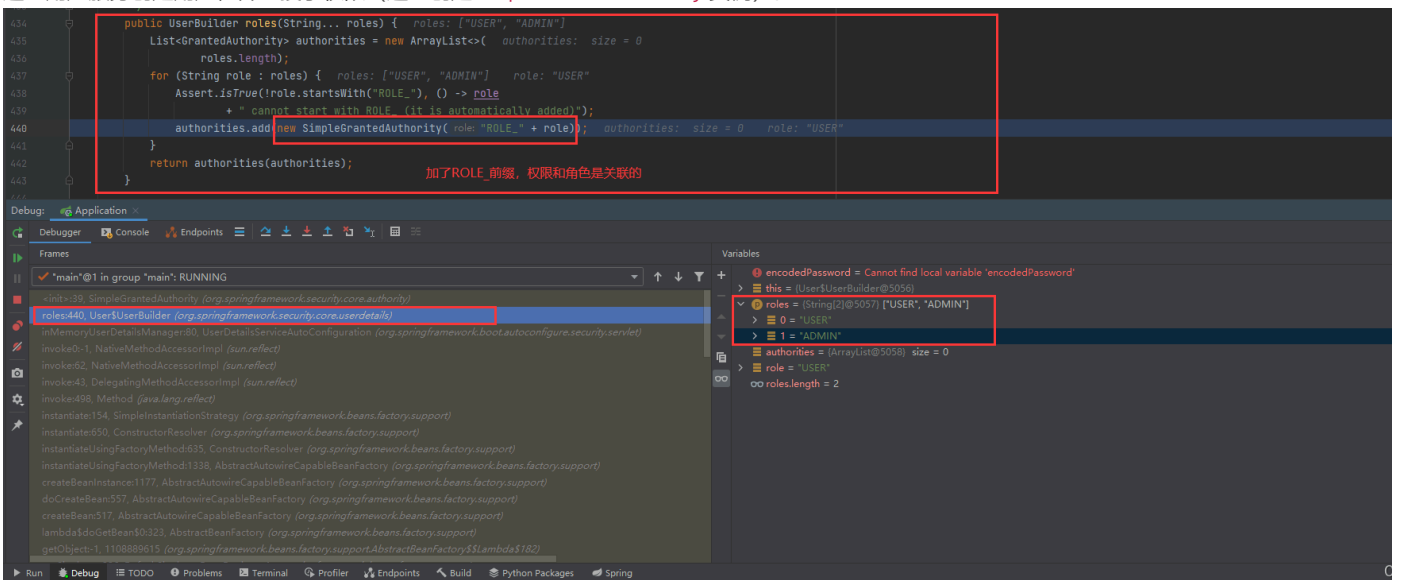


专栏

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。



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



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;
9     // 授权人的主体
10    private final Principal principal;
11 }
```

AuthorityGranter 接口源码：

```
1 /**
2  * AuthorityGranter接口用于将给定的主体
3  */
```



ITKaven

关注

11

1

27

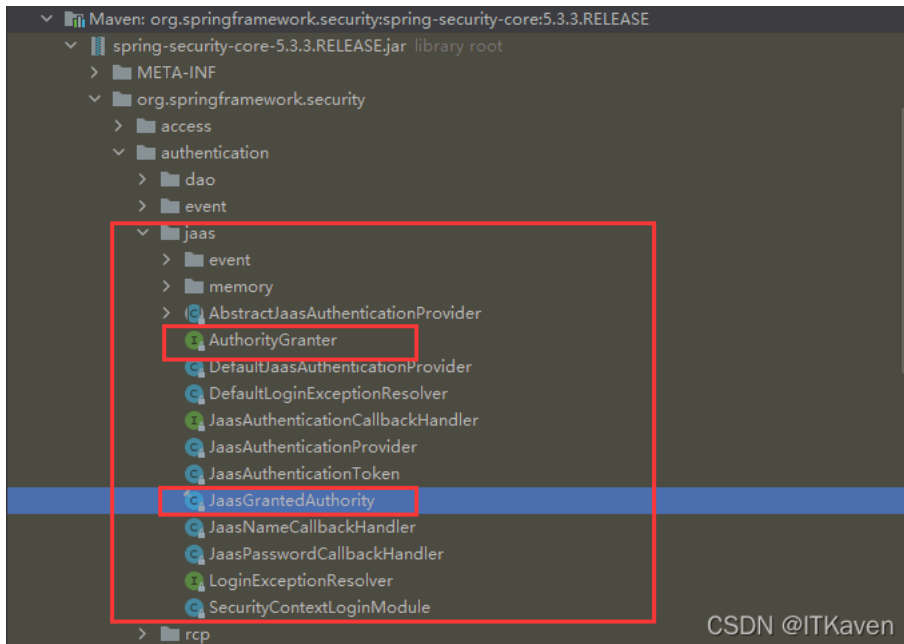
1

1

1

专栏

```
4  */
5  public interface AuthorityGranter {
6      /**
7       * 根据主体映射到角色名称集合
8       */
9      Set<String> grant(Principal principal);
10 }
```



`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，感谢 JAAS。

SwitchUserGrantedAuthority

`SwitchUserGrantedAuthority` 类源码：

```
1  /**
2   * SwitchUserFilter使用的自定义GrantedAuthority
3   * 存储原始用户的Authentication对象，以便从退出用户切换时使用。
4   */
5  public final class SwitchUserGrantedAuthority implements GrantedAuthority {
6
7      private static final long serialVersionUID = SpringSecurityCoreVersion.SERIAL_VERSION_UID;
8
9      private final String role;
10     // 存储原始用户的Authentication对象
11     private final Authentication source;
```

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 `kaven`（`root` 用户有这个权限）。

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



ITKaven

关注

11

1

27

1

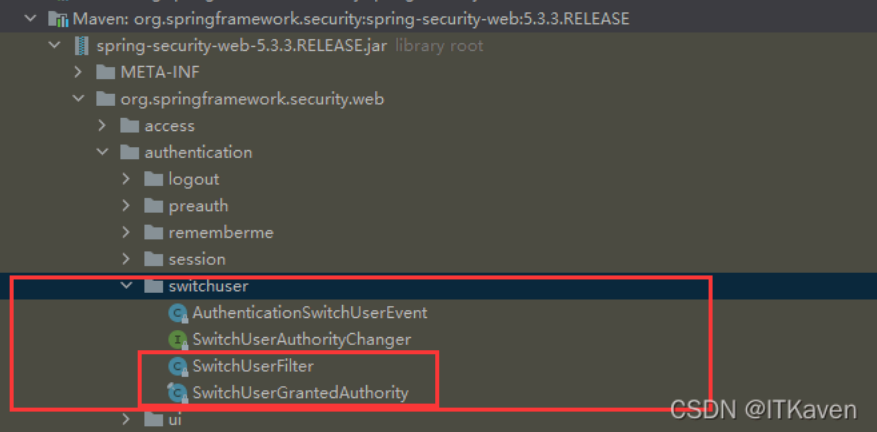
1

1

专栏



SwitchUserGrantedAuthority 是 SwitchUserFilter 使用的自定义 GrantedAuthority , SwitchUserFilter (用户切换处理过滤器) 负责用户上下文切换 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) 切换到普通用户 (如 ROLE_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 使用户切换的退出。

```
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; import org.springframework.security.core.GrantedAuthority; import org.springframework.security.core.userdetails.UserDetails; import java.util.ArrayL

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

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

【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_securitycontextholder-CSDN...

String getAuthority(); //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



ITKaven

关注

11



27



1



专栏



私信

关注



搜博主文章



热门文章

- MyBatis-Plus 之分页查询 167673
- 同步和异步的区别 89904
- 怎么保存退出 vim 编辑 70673
- MySQL的driverClassName、url 50048
- Spring 中 ClassPathXmlApplicationContext 类的简单使用 40177

分类专栏

- JAVA 30篇
- Spring 9篇
- Spring Boot 31篇
- Spring Security 12篇
- Spring Cloud 12篇
- Spring Cloud Alibaba 12篇

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环 ...
- ElasticJob-Lite：作业监听器
AKKO111: 那感觉没必要用这个Distribute的listener了，直接用ElasticJobServiceLo ...
- 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



ITKaven

关注



11



27



1



专栏

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



ITKaven

关注



11



27



1



专栏