角色权限判断

除了之前讲解的内置权限控制。Spring Security 中还支持很多其他权限控制。这些方法一般都用于用户已经被认证后，判断用户是否具有特定的要求。

hasAuthority(String)

判断用户是否具有特定的权限，用户的权限是在自定义登录逻辑中创建 User 对象时指定的。下图中 admin和normal 就是用户的权限。admin和normal 严格区分大小写。

```
return new User(username,password, AuthorityUtils.commaSeparatedStringToAuthorityList( authorityString: "admin,normal"));
```

在配置类中通过 hasAuthority("admin")设置具有 admin 权限时才能访问。

```
.antMatchers("/main1.html").hasAuthority("admin")
```

hasAnyAuthority(String ...)

如果用户具备给定权限中某一个，就允许访问。

下面代码中由于大小写和用户的权限不相同，所以用户无权访问

```
.antMatchers("/main1.html").hasAnyAuthority("adMin","admiN")
```

hasRole(String)

如果用户具备给定角色就允许访问。否则出现 403。

参数取值来源于自定义登录逻辑 `UserDetailsService` 实现类中创建 User 对象时给 User 赋予的授权。

在给用户赋予角色时角色需要以：`ROLE_开头`，后面添加角色名称。例如：ROLE_abc 其中 abc 是角色名，ROLE_是固定的字符开头。

使用 hasRole()时参数也只写 abc 即可。否则启动报错。

给用户赋予角色：

```
return new User(username,password, AuthorityUtils.commaSeparatedStringToAuthorityList( authorityString: "admin,normal,
ROLE_abc"));
```

在配置类中直接写 abc 即可。

```
.antMatchers("/main1.html").hasRole("abc")
```