文章编号：2096-1472(2020)-12-24-03
DOI:10.19644/j.cnki.issn2096-1472.2020.12.007

基于前后端分离架构的用户权限控制系统设计与实现

廖 祥

(四川轻化工大学自动化与信息工程学院，四川 自贡 643000)
✉893422451@qq.com

摘 要：本文基于前后端分离的架构，设计了多用户企业管理系统中用户角色权限控制系统，实现了不同用户基于角色的访问控制。在本系统中前端框架主要使用了Vue.js，后端则使用了Spring Boot进行快速构建，同时使用了Spring Security作为整个系统的安全框架。根据企业管理系统的需求实现了角色权限管理功能，以及用户的动态权限分配，同时兼顾了系统对可扩展性的需求。

关键词：角色权限控制；前后端分离架构；企业管理系统
中图分类号：TP311 **文献标识码**：A

Design and Implementation of User Permission Control System in a Separated Front-end and Rear-end Architecture

LIAO Xiang

(*School of Automation and Information Engineering*, *Sichuan University of Science & Engineering*, *Zigong* 643000, *China*)
✉893422451@qq.com

Abstract: This paper proposes a user permission control system based on structure of front and back end separation in multi-user enterprise management system. It helps to realize the role-based access control of different users. In this system, the front-end framework mainly uses Vue.js, and the back-end uses the Spring Boot for rapid construction. At the same time, Spring Security is used as the security framework of the entire system. According to the requirements of the enterprise management system, role authority management and users' dynamic permission are realized, and the system's demand for extensibility is also taken into account.

Keywords: role permission control; front and rear end separation; enterprise management system

1 引言(Introduction)

随着互联网应用架构的不断演变，前后端分离的开发模式已经成为当前一种流行的趋势[1]。后端不再控制页面如何进行展示，只需要对外暴露接口即可，而页面的渲染与展示全部由前端来完成，前端只需要调用后端服务的接口来提交和获取相应的数据即可。虽然采用前后端分离的开发模式具有很多优势，但是在页面权限和后端接口权限的控制上也带来了一些挑战[2]。传统模式下开发的Web系统前后端并不是独立工作，在判断某个用户是否拥有某一权限的时候，大多采用基于Spring AOP提供的面向切面编程的技术[3]。而在前后端分离的情况下，后端只对外提供接口，前端并不能直接调用后端系统中其他业务逻辑代码，只能通过调用接口获取数据来进行权限的判断，因此页面所展现的内容应该在用户登录之后根据用户的角色获取相关的资源数据后进行渲染。

2 系统技术概述(System technology overview)

由于采用的是前后端分离的架构，本文选择了当前互联网应用开发中比较流行的前后端框架：Vue.js和Spring Boot。数据库采用MySql5.7，持久层框架选用MyBatis，并且使用Spring Security来控制用户访问资源的权限并保证后端接口的安全性。

2.1 Vue.js

Vue.js是一套用于构建用户界面的渐进式JavaScript框

架，它将整个页面的内容都划分成组件的形式，每个组件都拥有各自的CSS样式和JavaScript代码，并且能够在其他组件中复用[4]。Vue.js除了能简化开发，它还提供了路由守卫的功能，在路由跳转的时候，可以进行一系列的逻辑判断，便于页面权限的控制。

2.2 Spring Boot

Spring框架的出现解决了传统EJB开发繁重臃肿的问题，为J2EE提供了一套简单实用的解决方案[5]。但是随着互联网应用功能越来越多，系统越来越复杂，大量的XML配置以及与第三方框架的整合工作让开发者痛苦不已。为了解决这些烦琐的配置和整合问题，Spring Boot应运而生，该框架采用了特定的方式进行自动配置，对于第三方框架的整合，也只需要导入相关依赖，使用少量的注解进行配置即可使用。

2.3 Spring Security

Spring Security基于Spring框架，为Web应用的安全提供了一套完整的解决方案[6]，该框架通过一系列不同功能的过滤器所组成的过滤器链来进行用户身份的认证和，同时也在Spring Boot中进行了整合，使用的时候只需要在Maven项目中导入依赖即可。

3 系统关键功能的实现(The realization of key functions of the system)

3.1 数据库表的设计

本文数据库表采用基于角色的访问控制(RBAC)来设计，主要包括了：用户表(hr)、用户角色关联表(hr_role)、角色表(role)、角色资源关联表(role_menu)和资源表(menu)。表之间的对应关系如图1所示。

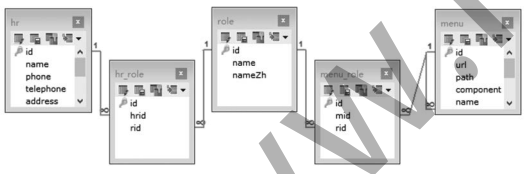

图1 数据库表对应关系

Fig.1 The corresponding relationship of database tables

其中用户表和角色表的对应关系为一对多，表示一个用户可以有多个角色；角色表与资源表的对应关系同样为一对多，表示一个角色可以拥有多个资源的访问权限，其中资源表的部分字段如表1所示。

表1 资源表部分字段

Tab.1 Partial fields of the resource table

列名	数据类型	长度
id	int	16
url	varchar	64
path	varchar	64
component	varchar	64
name	varchar	64

(1)url字段表示后端接口的请求路径，该字段主要用于后台接口的权限校验。例如“/employee/**”表示匹配员工管理功能模块的所有请求，用户在员工管理页面进行操作时，向后端发送的请求都带有“/employee”前缀。

(2)path字段表示前端路由路径，该字段用于页面权限的校验。前端页面的跳转主要通过改变路由路径来实现，而一个路由路径对应一个组件，例如/emp/basic路径表示跳转到用户基础信息管理。

(3)component字段表示当前路由路径下要加载的组件，在路由信息发生改变的时候，会通过path字段找到对应的组件，在页面进行渲染。

(4)name字段表示左侧菜单的名字，在用户登录成功之后，将会获取用户能够访问的资源，即菜单列表进行渲染，同时路由跳转也是通过点击菜单来实现。

3.2 前端功能实现

管理系统主要分为登录页和主页两个页面，用户登录成功之后就跳转到主页。在进入主页之前调用后端接口获取用户角色所对应的资源，即左侧菜单列表，然后通过点击左侧菜单列表进行路由跳转，加载不同的功能页面，完成相关业务操作。

现在考虑另一种情况，某个用户登录之后记下了主页的地址，或者记下了一个自己没有权限访问的路由路径，然后在地址栏手动输入之后进行访问，此时系统就会出现安全隐患。因为是前后端分离的项目，前端路由变化引起的页面跳转后端并不知道，所以除了后端需要传递给前端特定角色的资源数据以外，前端同样需要做页面权限的校验。解决此问题的办法就是使用Vue.js的路由组件提供的路由守卫来进行逻辑判断，常用的路由守卫如表2所示。

表2 常用的路由守卫

Tab.2 Common routing guards

名称	钩子函数	调用时机
全局前置守卫	router.beforeEach(to from next)	路由跳转之前执行
全局后置守卫	router.afterEach(to from)	路由跳转之后执行
组件内前置守卫	beforeRouteEnter(to from next)	进入组件前执行
组件内后置守卫	beforeRouteLeave(to from)	进入组件后执行

路由钩子函数参数说明：

(1)to：即将要进入的目标路由对象。

(2)from：当前导航正要离开的路由。

(3)next：调用该方法来进行逻辑判断。直接执行next()正常跳转到to指定的路由路径；如果执行next()返回了false，则中断当前跳转，回到from指定的路由路径；另外在next方法中还可以执行特定的路由跳转，如直接调用next(/loginPage)表示跳转到登录页。

本系统中使用了全局前置守卫，在页面发生跳转的时候通过在next方法中调用后端接口，来判断当前用户是否有权访问to所指定的页面，整个控制流程如图2所示。

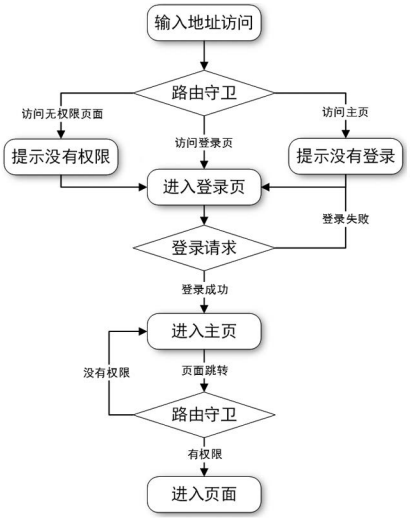

图2 前端页面权限控制流程

Fig.2 Front-end page permission control process

3.3 后端功能实现

使用Spring Boot脚手架快速搭建好项目，完成基础数据库表数据的增删改查后，在Maven项目的pom文件中导入Spring Security的依赖：

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

导入依赖后，关键功能的实现过程主要分为如下几个步骤：

(1)用户类Hr实现UserDetails接口。UserDetails接口是Spring Security提供的一个与用户信息密切相关的接口，后台管理系统中的用户实体类实现了该接口并重写其接口方法之后，Spring Security框架就能够获取业务系统中的用户信息，并基于此信息来完成认证与授权。

(2)用户业务类HrService实现UserDetailsService接口。该接口只提供了loadUserByUsername一个方法，在重写该方法后通过用户名查询用户表返回用户数据对象即可。

(3)自定义FilterInvocationSecurityMetadataSource接口实现类。该实现类作为一个资源管理器，主要目的就是根据前端请求的路径，查询哪些角色拥有该资源的访问权限，这些信息将从角色资源表及其关联表中查询得到。

(4)自定义AccessDecisionManager接口实现类。该实现类作为决策管理器，接口方法的功能主要是从用户成功登录时保存的用户信息中获取到用户角色，然后判断当前用户的角色是否在资源管理器所查询出角色集合中，如果角色集合中包含当前用户的角色，则表示当前用户有权访问。

(5)在完成上面的工作之后，还需要对Spring Security框架进行配置。Spring Security提供了WebSecurityConfigurerAdapter接口，实现该接口之后配置后端哪些接口需要进行拦截和权限校验，并且指定校验成功和失败的处理器。完成上面的配置之后，后端接口的权限校验功能就已经实现了，整个校验流程如图3所示。

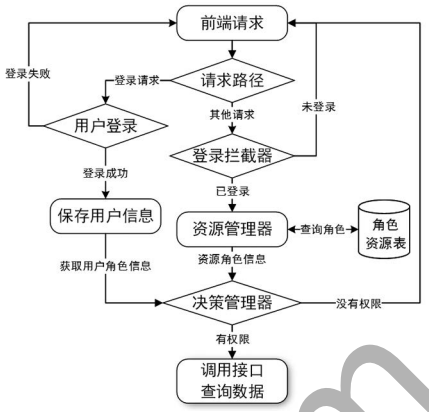

图3 后端接口权限校验流程

Fig.3 Back-end interface permission verification process

3.4 系统功能测试

首先使用PostMan接口调试工具测试后台接口的权限，在未登录的情况，直接调用其他接口获取数据均不成功，返回了未登录的提示信息，登录之后成功访问。在访问没有权限的接口时，请求被拦截，返回未授权提示信息。通过系统管理员添加测试用户并分配相应的角色，登录进入主页之后只看到了当前角色能够访问的菜单列表，手动输入其他路由路径显示未授权的提示信息。

4 结论(Conclusion)

本文基于当前流行的前后端分离架构，设计实现了在此架构下用户角色权限控制系统，从对后端接口的测试以及前后端联调的运行结果来看，此系统能够成功地控制页面权限与后台接口权限，可以满足企业级后台管理系统的需要。如果后期系统在进行功能扩展时前端加入了新的页面，运维人员只需要在数据库资源表添加新增页面相关数据即可，后端权限相关代码可以做到零修改，因此该系统在做到安全、高效的前提下同时也兼顾了整个系统的可扩展性。

参考文献(References)

[1] 王建,罗政,张希,等.Web项目前后端分离的设计与实现[J].软件工程,2020,23(04):22-24.

[2] 迟殿委.前后端分离的Web架构解决方案[J].智慧工厂,2019(06):37-38.

[3] 宋清卿.前后端分离Web系统下一种访问控制方法的设计与实现[J].计算机仿真,2020(05):23-26.

[4] 刘金羽.基于Vue.js的前端教学软件设计与实现[J].电脑编程技巧与维护,2020(02):23-24;29.

[5] Zhang M, Lv J, Jiang Y, et al. Intelligent business cloud service platform based on SpringBoot framework[C]. 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC). IEEE, 2020:201-207.

[6] 孙云杰,段祎林,田盼栋,等.基于SSM+Spring Security的多角色登录功能后台实现[J].电子技术与软件工程,2020(05):16-18.

作者简介：

廖 祥(1994-)，男，硕士生.研究领域：网络化控制系统，软件系统开发.