1 实验10 SpringBoot集成Spring Security

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 1910705127 | 姓名 | 宋宇昕 | 班级 | 19软件1班 |
| 上课地点 | 精工楼 | 教师 | 梦昭睿 | 上课时间 | 11.18 |

UserService文件

**package com.example.springboot.service;**

**import com.example.springboot.entity.SysUser;**

**import com.example.springboot.mapper.SysUserMapper;**

**import org.springframework.beans.factory.annotation.Autowired;**

**import org.springframework.stereotype.Service;**

**@Service**

**public class SysUserService {**

**@Autowired**

**private SysUserMapper userMapper;**

**public SysUser selectById(Integer id) {**

**return userMapper.selectById(id);**

**}**

**public SysUser selectByName(String name) {**

**return userMapper.selectByName(name);**

**}**

**}**

用户权限

CREATE TABLE `sys\_user\_role` (

`user\_id` int(11) NOT NULL,

`role\_id` int(11) NOT NULL,

PRIMARY KEY (`user\_id`,`role\_id`),

KEY `fk\_role\_id` (`role\_id`),

CONSTRAINT `fk\_role\_id` FOREIGN KEY (`role\_id`) REFERENCES `sys\_role` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT `fk\_user\_id` FOREIGN KEY (`user\_id`) REFERENCES `sys\_user` (`id`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;



package com.example.springboot.securtiy;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.builders.WebSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration

@EnableWebSecurity

@EnableGlobalMethodSecurity(prePostEnabled = true)

public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

@Autowired

private CustomUserDetailsService userDetailsService;

@Override

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.userDetailsService(userDetailsService).passwordEncoder(new PasswordEncoder() {

@Override

public String encode(CharSequence charSequence) {

return charSequence.toString();

}

@Override

public boolean matches(CharSequence charSequence, String s) {

return s.equals(charSequence.toString());

}

});

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

// 如果有允许匿名的url，填在下面

// .antMatchers().permitAll()

.anyRequest().authenticated()

.and()

// 设置登陆页

.formLogin().loginPage("/login")

// 设置登陆成功页

.defaultSuccessUrl("/").permitAll()

// 自定义登陆用户名和密码参数，默认为username和password

// .usernameParameter("username")

// .passwordParameter("password")

.and()

.logout().permitAll();

// 关闭CSRF跨域

http.csrf().disable();

}

@Override

public void configure(WebSecurity web) throws Exception {

// 设置拦截忽略文件夹，可以对静态资源放行

web.ignoring().antMatchers("/css/\*\*", "/js/\*\*");

}

}

