权限管理

可以使用 Servlet Filter ,也可以Spring MVC action拦截器实现注解式权限验证

public interface HandlerInterceptor {

boolean preHandle(

//执行action里面的处理逻辑之前执行 true 继续执行postHandle,afterCompletion

HttpServletRequest request, HttpServletResponse response,

Object handler)

throws Exception;

void postHandle(

// postHandle 在执行action里面的逻辑后返回视图执行

HttpServletRequest request, HttpServletResponse response,

Object handler, ModelAndView modelAndView)

throws Exception;

void afterCompletion( //在action返回视图后执行

HttpServletRequest request, HttpServletResponse response,

Object handler, Exception ex)

throws Exception;

}

添加自定义注解

@Documented

@Inherited

@Target(ElementType.METHOD)

@Retention(RetentionPolicy.RUNTIME)

public @interface AuthPassport {

boolean validate() default true;

}

添加自己的拦截器实现AuthInterceptor继承于HandlerInterceptorAdapter

public class AuthInterceptor extends HandlerInterceptorAdapter {

@Override

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) throws Exception {

if(handler.getClass().isAssignableFrom(HandlerMethod.class)){

AuthPassport authPassport = ((HandlerMethod) handler).getMethodAnnotation(AuthPassport.class);

//没有声明需要权限,或者声明不验证权限

if(authPassport == null || authPassport.validate() == false)

return true;

else{

//在这里实现自己的权限验证逻辑

if(false)//如果验证成功返回true（这里直接写false来模拟验证失败的处理）

return true;

else//如果验证失败

{

//返回到登录界面

response.sendRedirect("account/login");

return false;

}

}

}

else

return true;

}

}

<mvc:interceptors>

<!-- 国际化操作拦截器 如果采用基于（请求/Session/Cookie）则必需配置 -->

<bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor" />

<!-- 如果不定义 mvc:mapping path 将拦截所有的URL请求 -->

<bean class="com.demo.web.auth.AuthInterceptor"></bean>

</mvc:interceptors>

这样在执行每个action方法是都会调用AuthInterceptor处理，当判断action上有我们定义AuthPassport注解时就会执行里面的权限验证逻辑。

@AuthPassport

@RequestMapping(value={"/index","/hello"})

public ModelAndView index(){

ModelAndView modelAndView = new ModelAndView();

modelAndView.addObject("message", "Hello World!");

modelAndView.setViewName("index");

return modelAndView;

}