18、Spring Boot与安全

2020年8月17日 15:42

- 一、安全

  Spring Security是针对Spring项目的安全框架，也是Spring Boot底层安全模块默认的技术选型。他可以实现强大的web安全控制。对于安全控制，我们仅需引入spring-boot-starter-security模块，进行少量的配置，即可实现强大的安全管理。

  几个类：

  WebSecurityConfigurerAdapter：自定义Security策略

  AuthenticationManagerBuilder：自定义认证策略

  @EnableWebSecurity：开启WebSecurity模式

  - 应用程序的两个主要区域是“认证”和“授权”（或者访问控制）。这两个主要区域是Spring Security的两个目标。
  - “认证”（Authentication），是建立一个他声明的主体的过程（一个“主体”一般是指用户，设备或一些可以在你的应用程序中执行动作的其他系统）。
  - “授权”（Authorization）指确定一个主体是否允许在你的应用程序执行一个动作的过程。为了抵达需要授权的店，主体的身份已经有认证过程建立。
  - 这个概念是通用的而不只在Spring Security中。

- 二、Web&安全

  - 导入依赖

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>
```

  - 整体代码

    config下的

```
package com.lhq.security.config;
import
org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerB
uilder;
```

```
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapte
r;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
@EnableWebSecurity
public class MySecurityConfig extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        //定制请求功能
        http.authorizeRequests().antMatchers("/").permitAll()
                .antMatchers("/level1/**").hasRole("VIP1")
                .antMatchers("/level2/**").hasRole("VIP2")
                .antMatchers("/level3/**").hasRole("VIP3");
        //开启自动配置的登录功能
        http.formLogin().usernameParameter("username").passwordParameter("password").
                loginPage("/userlogin");
        //开启自动配置的注销功能
        http.logout().logoutSuccessUrl("/");
        //开启记住我功能
//      http.rememberMe().rememberMeParameter("rem");
    }

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
//      super.configure(auth);
        auth.inMemoryAuthentication().passwordEncoder(new
BCryptPasswordEncoder()).withUser("zhangsan")
                .password(new
BCryptPasswordEncoder().encode("123456")).roles("VIP1","VIP2");
        auth.inMemoryAuthentication().passwordEncoder(new
BCryptPasswordEncoder()).withUser("lisi")
                .password(new
BCryptPasswordEncoder().encode("123456")).roles("VIP2","VIP3");
        auth.inMemoryAuthentication().passwordEncoder(new
BCryptPasswordEncoder()).withUser("wangwu")
                .password(new
BCryptPasswordEncoder().encode("123456")).roles("VIP1","VIP2","VIP3");
    }
```

html下的

```
//添加sec标签支持
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
//判断是否登录
<div sec:authorize="!isAuthenticated()">
  <h2 align="center">游客您好，如果想查看武林秘籍 <a th:href="@{/userlogin}">请登录</a>
```

```
</h2>
</div>
<div sec:authorize="isAuthenticated()">
  <h2><span sec:authentication="name"></span>，您好，您的角色有:
    <span sec:authentication="principal.authorities"></span> </h2>
</div>
//登录角色权限判断
<div sec:authorize="hasRole('VIP1')">
  <h3>普通武功秘籍</h3>
  <ul>
    <li><a th:href="@{/level1/1}">罗汉拳</a></li>
    <li><a th:href="@{/level1/2}">武当长拳</a></li>
    <li><a th:href="@{/level1/3}">全真剑法</a></li>
  </ul>
</div>
<div sec:authorize="hasRole('VIP2')">
  <h3>高级武功秘籍</h3>
  <ul>
    <li><a th:href="@{/level2/1}">太极拳</a></li>
    <li><a th:href="@{/level2/2}">七伤拳</a></li>
    <li><a th:href="@{/level2/3}">梯云纵</a></li>
  </ul>
</div>
<div sec:authorize="hasRole('VIP3')">
  <h3>绝世武功秘籍</h3>
  <ul>
    <li><a th:href="@{/level3/1}">葵花宝典</a></li>
    <li><a th:href="@{/level3/2}">龟派气功</a></li>
    <li><a th:href="@{/level3/3}">独孤九剑</a></li>
  </ul>
```

- 登陆/注销
  - HttpSecurity配置登陆、注销功能
  - 登录

    ```
    //config
    //自定义登录页面
    http.formLogin().usernameParameter("username").passwordParameter("password").
                            loginPage("/userlogin");
    //controller
    private final String PREFIX = "pages/";
    @GetMapping("/userlogin")
    public String loginPage() {
       return PREFIX+"login";
    }
    ```
  - 注销

    ```
    //config
    ```

```
//开启自动配置的注销功能
http.logout().logoutSuccessUrl("/");
//html发送logout自动退出登录
<form th:action="@{/logout}" method="post">
   <input type="submit" value="注销">
</form>
```

- Thymeleaf提供的SpringSecurity标签支持
  //添加sec标签支持
  ```
  <html xmlns:th="http://www.thymeleaf.org"
        xmlns:sec="http://www.thymeleaf.org/extras/spring-security">
  ```
  - 需要引入thymeleaf-extras-springsecurity5
  - sec:authentication="name"获得当前用户的用户名
    ```
    <h2><span sec:authentication="name"></span>，您好，您的角色有：
        <span sec:authentication="principal.authorities"></span>
    </h2>
    ```
  - sec:authorize="hasRole('ADMIN')"当前用户必须拥有ADMIN权限时才会显示标签内容
    ```
    //登录角色权限判断
    <div sec:authorize="hasRole('VIP1')">
       <h3>普通武功秘籍</h3>
       <ul>
          <li><a th:href="@{/level1/1}">罗汉拳</a></li>
          <li><a th:href="@{/level1/2}">武当长拳</a></li>
          <li><a th:href="@{/level1/3}">全真剑法</a></li>
       </ul>
    </div>
    ```
- remember me
  - 表单添加remember-me的checkbox
    ```
    <input type="checkbox" name="rem">记住我
    ```
  - 配置启用remember-me功能
    ```
    //开启记住我功能
    http.rememberMe().rememberMeParameter("rem");
    ```
- CSRF（Cross-site request forgery）跨站请求伪造
  HttpSecurity启用csrf功能，会为表单添加_csrf的值，提交携带来预防CSRF；