

Spring Security使用自定义控制器来完成登陆验证



作者 叶飞 (/u/b3be0432b2e3) [+ 关注](#)

2016.07.08 00 : 38 * 字数1288 阅读729 评论1 喜欢3

(/u/b3be0432b2e3)

Spring Security下面简称为Security基于spring-security 4.1

安全的WEB扩展中形式方式登陆使用的是过滤器方式，页面模版是可以定制的，但是如果需要登陆表单中有更多的选项，或者说需要在登陆的时候处理一些事情就变的很不方便。下面就是教你如何使用一个普通的控制器来完成登陆验证。

首先是制定安全的配置文件

在http节点中有一个属性**entry-point-ref**可以指定如果需要登陆将如何反应，在实现**AuthenticationEntryPoint**接口的类中有一个名称叫**LoginUrlAuthenticationEntryPoint**的类他可以实现需要登陆的时候产生URL跳转。

首先创建一个LoginUrlAuthenticationEntryPoint bean

```
<beans:bean id="authenticationEntryPoint" class="org.springframework.security.web.authentication
    <!-- 构造函数中指定需要跳转的 URL -->
    <beans:constructor-arg value="/login" />
</beans:bean>
```

然后在http节点中配置**entry-point-ref**
auto-config可以关闭，因为不需要自动配置

```
<http auto-config="false" entry-point-ref="authenticationEntryPoint">
    ...
</http>
```

在http节点内添加intercept-url防止拦截/登录链接

```
<intercept-url pattern="/login" access="permitAll" />
```

如果需要记住我功能，需要在HTTP节点内增加**记得，我的配置**

```
<!-- rememberMe 对应的是表单类中的属性名称 -->
<remember-me remember-me-parameter="rememberMe" />
```

最后可以增加一个退出过滤器

因为是过滤器拦截判断使用/注销不需要有对应的控制器

如果这个/注销中没有对应的控制器，需要添加**logout-success-url**跳转链接，防止访问/注销URL后候报404错误

```
<logout logout-url="/logout" logout-success-url="/" />
```

最后需要配置**认证管理器**节点中的ID用来给控制器中注入使用

```
<authentication-manager id="authenticationManager">
    ...
</authentication-manager>
```

完整的配置文件代码



```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/security"
  xmlns:beans="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/security
    http://www.springframework.org/schema/security/spring-security.xsd">

  <!-- 本配置文件完全使用自定的控制器来完成登陆退出操作 Yefei -->

  <!-- 开启 Spring Security 调试模式
  <debug />
  -->

  <!-- 是否开启注解支持, 例如: @Secured
  <global-method-security secured-annotations="enabled" />
  -->

  <!-- 配置不需要安全过滤的页面 -->
  <http pattern="/static/**" security="none" />

  <beans:bean id="authenticationEntryPoint"
    class="org.springframework.security.web.authentication.LoginUrlAuthenticationEntryPoint"
    <beans:constructor-arg value="/login" />
  </beans:bean>

  <http auto-config="false"
    entry-point-ref="authenticationEntryPoint">
    <intercept-url pattern="/" access="permitAll"/>
    <intercept-url pattern="/login" access="permitAll" />
    <intercept-url pattern="/admin/**" access="hasRole('ADMIN')"/>
    <intercept-url pattern="/**" access="hasRole('USER')"/>
    <!-- 用于 cookie 登陆 remember-me-parameter 中的值必须和表单中的 rememberMe name 一致 -->
    <remember-me remember-me-parameter="rememberMe" />
    <!-- logout 可以使用简单的过滤器完成, 启用了 CSRF 必须使用 POST 退出 -->
    <logout logout-url="/logout" logout-success-url="/" />
  </http>

  <authentication-manager id="authenticationManager">
    <authentication-provider>
      <user-service>
        <user name="admin" authorities="ROLE_USER,ROLE_ADMIN" password="123456" />
        <user name="user" authorities="ROLE_USER" password="123456" />
      </user-service>
    </authentication-provider>
  </authentication-manager>

</beans:beans>
```



下面是控制器的代码部分

代码过程都是通过debug模式下在**UsernamePasswordAuthenticationFilter**中分析得出



```
@Controller
public class AuthController {

    @Autowired
    @Qualifier("authenticationManager") // bean id 在 <authentication-manager> 中设置
    private AuthenticationManager authManager;

    @Autowired
    private SessionAuthenticationStrategy sessionStrategy;

    @Autowired(required = false)
    private RememberMeServices rememberMeServices;

    @Autowired(required = false)
    private ApplicationEventPublisher eventPublisher;

    @RequestMapping(value = "/login", method = RequestMethod.GET)
    public String login(LoginForm form) {
        return "login";
    }

    @RequestMapping(value = "/login", method = RequestMethod.POST)
    public String loginPost(@Valid LoginForm form, BindingResult result,
        HttpServletRequest request, HttpServletResponse response) {
        if (!result.hasErrors()) {

            // 创建一个用户名密码登陆信息
            UsernamePasswordAuthenticationToken token =
                new UsernamePasswordAuthenticationToken(form.getUsername(), form.getPassword());

            try {
                // 用户名密码登陆效验
                Authentication authResult = authManager.authenticate(token);

                // 在 session 中保存 authResult
                sessionStrategy.onAuthentication(authResult, request, response);

                // 在当前访问线程中设置 authResult
                SecurityContextHolder.getContext().setAuthentication(authResult);

                // 如果记住我在配置文件中有配置
                if (rememberMeServices != null) {
                    rememberMeServices.loginSuccess(request, response, authResult);
                }

                // 发布登陆成功事件
                if (eventPublisher != null) {
                    eventPublisher.publishEvent(new InteractiveAuthenticationSuccessEvent(authResult));
                }

                return "redirect:/";
            }
        }
    }
}
```



```
    } catch (AuthenticationException e) {
        result.reject("authentication.exception", e.getLocalizedMessage());
    }
}
return "login";
}
```

最后是JSP页面部分代码

页面都需要经过安全过滤器才能产生csrf令牌，否则请关闭csrf

登陆代码

```
<form:form commandName="loginForm" method="POST">
    <form:errors path="" />
    <p>用户名: <form:input path="username"/> <form:errors path="username" /></p>
    <p>密码: <form:input path="password"/> <form:errors path="password" /></p>
    <p>记住我: <form:checkbox path="rememberMe"/> <form:errors path="rememberMe" /></p>
    <button type="submit">登陆</button>
</form:form>
```

退出代码

```
<form action="/logout" method="POST">
    <sec:csrfInput/>
    <button>退出</button>
</form>
```

原创文章转载注明



叶飞 (/u/b3be0432b2e3)

写了1288字，被1人关注，获得了3个喜欢

(/u/b3be0432b2e3)

+ 关注

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

赞赏支持



(/u/716db6199928)

喜欢 | 3



更多分享

(http://cwb.assets.jianshu.io/notes/images/4707133/weibo/image_1



写下你的评论...

1评论条

只看作者

喜欢按排序 按时间序正 按时间倒序



磊客 (/u/716db6199928)

2楼 · 2016.07.08 09:39

(/u/716db6199928)

支持



赞



回复



被以下专题收入，发现更多相似内容

