

springboot整合shiro-实现验证码认证(九)

原文地址，转载请注明出处：https://blog.csdn.net/qq_34021712/article/details/80470738 ©王赛超

验证码是有效防止暴力破解的一种手段，常用做法是在服务端产生一串随机字符串与当前用户会话关联（我们通常说的放入 Session），然后向终端用户展现一张经过“扰乱”的图片，只有当用户输入的内容与服务端产生的内容相同时才允许进行下一步操作。

shiro 添加验证码

说的是shiro添加验证码,其实不如说是web服务登录功能添加验证码,因为这个功能和shiro完全没有任何关系,网上大都是实现 FormAuthenticationFilter 然后在filter中进行验证码校验,或者是 在shiroRealm中进行验证码校验,一大堆的代码要写,感觉很麻烦,下面进行最简单的代码实现验证码功能。直接在 login方法内,判断验证码是否正确

验证码工具类

```
1 package com.springboot.test.shiro.global.utils;
2
3 import java.awt.Color;
4 import java.awt.Font;
5 import java.awt.Graphics;
6 import java.awt.image.BufferedImage;
7 import java.io.File;
8 import java.io.FileOutputStream;
9 import java.util.Random;
10
11 import javax.imageio.ImageIO;
12
13 /**
14  * 验证码工具类
15  */
16 public class CaptchaUtil {
17
18     // 随机产生的字符串
19     private static final String RANDOM_STRS = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
20
21     private static final String FONT_NAME = "Fixedsys";
22     private static final int FONT_SIZE = 18;
23
24     private Random random = new Random();
25
26     private int width = 80; // 图片宽
27     private int height = 25; // 图片高
28     private int lineNum = 50; // 干扰线数量
29     private int strNum = 4; // 随机产生字符数量
30
31     /**
32      * 生成随机图片
33      */
34     public BufferedImage genRandomCodeImage(StringBuffer randomCode) {
35         // BufferedImage类是具有缓冲区的Image类
36         BufferedImage image = new BufferedImage(width, height,
37             BufferedImage.TYPE_INT_BGR);
38         // 获取Graphics对象,便于对图像进行各种绘制操作
39         Graphics g = image.getGraphics();
40         // 设置背景色
41         g.setColor(getRandColor(200, 250));
42         g.fillRect(0, 0, width, height);
43
44         // 设置干扰线的颜色
45         g.setColor(getRandColor(110, 120));
46
47         // 绘制干扰线
48         for (int i = 0; i <= lineNum; i++) {
49             drawLine(g);
50         }
51         // 绘制随机字符
52         g.setFont(new Font(FONT_NAME, Font.ROMAN_BASELINE, FONT_SIZE));
53         for (int i = 1; i <= strNum; i++) {
54             randomCode.append(drawString(g, i));
55         }
56         g.dispose();
57         return image;
58     }
59
60     /**
61      * 给定范围获得随机颜色
62      */
63     private Color getRandColor(int fc, int bc) {
```

```
64     if (fc > 255){
65         fc = 255;
66     }
67     if (bc > 255){
68         bc = 255;
69     }
70     int r = fc + random.nextInt(bc - fc);
71     int g = fc + random.nextInt(bc - fc);
72     int b = fc + random.nextInt(bc - fc);
73     return new Color(r, g, b);
74 }
75
76 /**
77  * 绘制字符串
78  */
79 private String drawString(Graphics g, int i) {
80     g.setColor(new Color(random.nextInt(101), random.nextInt(111), random
81         .nextInt(121)));
82     String rand = String.valueOf(getRandomString(random.nextInt(RANDOM_STRS
83         .length())));
84     g.translate(random.nextInt(3), random.nextInt(3));
85     g.drawString(rand, 13 * i, 16);
86     return rand;
87 }
88
89 /**
90  * 绘制干扰线
91  */
92 private void drawLine(Graphics g) {
93     int x = random.nextInt(width);
94     int y = random.nextInt(height);
95     int x0 = random.nextInt(16);
96     int y0 = random.nextInt(16);
97     g.drawLine(x, y, x + x0, y + y0);
98 }
99
100 /**
101  * 获取随机的字符
102  */
103 private String getRandomString(int num) {
104     return String.valueOf(RANDOM_STRS.charAt(num));
105 }
106
107 public static void main(String[] args) {
108     CaptchaUtil tool = new CaptchaUtil();
109     StringBuffer code = new StringBuffer();
110     BufferedImage image = tool.genRandomCodeImage(code);
111     System.out.println("random code = " + code);
112     try {
113         // 将内存中的图片通过流形式输出到客户端
114         ImageIO.write(image, "JPEG", new FileOutputStream(new File(
115             "/Users/wangsaichao/Desktop/random-code.jpg")));
116     } catch (Exception e) {
117         e.printStackTrace();
118     }
119 }
120 }
121 }
```

获取验证码Controller

```
1 package com.springboot.test.shiro.modules.login;
2
3 import com.springboot.test.shiro.global.utils.CaptchaUtil;
4 import org.springframework.stereotype.Controller;
5 import org.springframework.web.bind.annotation.RequestMapping;
6
7 import javax.imageio.ImageIO;
8 import javax.servlet.ServletException;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpSession;
12 import java.awt.image.BufferedImage;
13 import java.io.IOException;
14
15 /**
16  * @author: wangsaichao
17  * @date: 2018/5/26
```

```
18  * @description:
19  */
20  @Controller
21  public class CaptchaController {
22
23      public static final String KEY_CAPTCHA = "KEY_CAPTCHA";
24
25      @RequestMapping("/Captcha.jpg")
26      public void getCaptcha(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27          // 设置相应类型,告诉浏览器输出的内容为图片
28          response.setContentType("image/jpeg");
29          // 不缓存此内容
30          response.setHeader("Pragma", "No-cache");
31          response.setHeader("Cache-Control", "no-cache");
32          response.setDateHeader("Expires", 0);
33          try {
34
35              HttpSession session = request.getSession();
36
37              CaptchaUtil tool = new CaptchaUtil();
38              StringBuffer code = new StringBuffer();
39              BufferedImage image = tool.getRandomCodeImage(code);
40              session.removeAttribute(KEY_CAPTCHA);
41              session.setAttribute(KEY_CAPTCHA, code.toString());
42
43              // 将内存中的图片通过流形式输出到客户端
44              ImageIO.write(image, "JPEG", response.getOutputStream());
45
46          } catch (Exception e) {
47              e.printStackTrace();
48          }
49      }
50  }
51
52 }
```

在shiroConfig中对 获取验证码的功能开放权限

```
1  filterChainDefinitionMap.put("/Captcha.jpg", "anon");
```

在login方法内添加验证码校验

```
1  @RequestMapping(value = "/login", method = RequestMethod.POST)
2  public String loginUser(HttpServletRequest request, String username, String password, boolean rememberMe, String captcha, Model model) {
3
4      //校验验证码
5      //session中的验证码
6      String sessionCaptcha = (String) SecurityUtils.getSubject().getSession().getAttribute(CaptchaController.KEY_CAPTCHA);
7      if (null == captcha || !captcha.equalsIgnoreCase(sessionCaptcha)) {
8          model.addAttribute("msg", "验证码错误! ");
9          return "login";
10     }
11
12     //对密码进行加密
13     //password=new SimpleHash("md5", password, ByteSource.Util.bytes(username.toLowerCase() + "shiro"),2).toHex();
14     //如果有点击 记住我
15     UsernamePasswordToken usernamePasswordToken=new UsernamePasswordToken(username,password,rememberMe);
16     //UsernamePasswordToken usernamePasswordToken = new UsernamePasswordToken(username,password);
17     Subject subject = SecurityUtils.getSubject();
18     try {
19         //登录操作
20         subject.login(usernamePasswordToken);
21         return "redirect:index";
22     } catch (Exception e) {
23         //登录失败从request中获取shiro处理的异常信息 shiroLoginFailure:就是shiro异常类的全类名
24         String exception = (String) request.getAttribute("shiroLoginFailure");
25
26         if(e instanceof UnknownAccountException){
27             model.addAttribute("msg", "用户名或密码错误! ");
28         }
29
30         if(e instanceof IncorrectCredentialsException){
31             model.addAttribute("msg", "用户名或密码错误! ");
32         }
33
34         if(e instanceof LockedAccountException){
35             model.addAttribute("msg", "账号已被锁定,请联系管理员! ");
36         }
37     }
```

```
37
38     //返回登录页面
39     return "login";
40 }
41 }
```

登录页面添加验证码

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" xmlns:th="http://www.thymeleaf.org"
3     xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3"
4     xmlns:shiro="http://www.pollix.at/thymeleaf/shiro">
5 <head>
6     <meta charset="UTF-8" />
7     <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>欢迎登录</h1>
11 <h1 th:if="${msg != null}" th:text="${msg}" style="color: red"></h1>
12 <form action="/login" method="post">
13     用户名: <input type="text" name="username"/><br/>
14     密码: <input type="password" name="password"/><br/>
15     验证码: <input type="text" name="captcha"/>
16     (看不清<a href="javascript:void(0)" onclick="javascript:refreshCaptcha()">换一张</a>)<br/>
17     <input type="checkbox" name="rememberMe" />记住我<br/>
18     <input type="submit" value="提交"/> <button<a href="/unlockAccount">解锁admin用户</a></button>
19 </form>
20 </body>
21 <script type="text/javascript" th:src="@{/js/jquery.js}"></script>
22 <script type="text/javascript">
23     function kickout(){
24         var href=location.href;
25         if(href.indexOf("kickout")>0){
26             alert("您的账号在另一台设备上登录,如非本人操作,请立即修改密码! ");
27         }
28     }
29     window.onload=kickout();
30
31     function refreshCaptcha(){
32         $("#captcha_img").attr("src","/Captcha.jpg?id=" + new Date() + Math.floor(Math.random()*24));
33     }
34 </script>
35 </html>
```

测试结果

localhost:9090/login

欢迎登录

验证码错误！

用户名：

admin

密码：

.....

验证码：

xp18



(看不清[换一张](#))

☐

记住我

提交

[解锁admin用户](#)

https://blog.csdn.net/qq_34021712