验证码验证人工操作

- 验证码生成工具可以搜索开源网站, 或者自己写
- 感兴趣的同学可以自己开发术语自己的验证码工具
- 验证码验证思路就是 用户在登陆页面访问后台生成验证码接口, 并将真是code存入sesion
- 用户登陆的时候 拿用户输入的code 和 后台session里存储的 code 进行对比
- 通过则人工操作, 否则判断为机器人

```
String checkCode = req.getParameter( s: "checkCode");
Object code =   req.getSession( b: true).getAttribute( s: "result");

if ( checkCode == null || !checkCode.equals( code.toString() ) ) {
    req.getSession( b: true).setAttribute( s: "passworderror",  o: "验证码不正确");
    resp.sendRedirect( s: req.getContextPath()+ "/register.jsp");
    return ;
}
```

图形验证码类

```
package com.lovecoding.xml.study.utils;
import javax.imageio.ImageIO;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.util.Random;

public class Captcha {
    public static final String RANDOMCODEKEY = "result";//放到session中的key
    private Random random = new Random();
    private String randString = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghigklmnopqrstuvwxyz";//随机产生的字符串

    private int width = 80;//图片宽
    private int height = 26;//图片高
    private int lineSize = 100;//干扰线数量
```

```
18      private int stringNum = 4;//随机产生字符数量
19
20      /**
21       * 生成随机图片
22       */
23      public void getRandcode(HttpServletRequest request, HttpServletResponse response) {
24
25          response.setContentType("image/jpeg");
26
27          HttpSession session = request.getSession(true);
28          //BufferedImage类是具有缓冲区的Image类,Image类是用于描述图像信息的类
29          BufferedImage image = new
    BufferedImage(width,height,BufferedImage.TYPE_INT_BGR);
30          //产生Image对象的Graphics对象,改对象可以在图像上进行各种绘制操作
31          Graphics g = image.getGraphics();
32          g.fillRect(0, 0, width, height);
33          g.setFont(new Font("Times New Roman",Font.ROMAN_BASELINE,18));
34          g.setColor(getRandColor(160, 200));
35          //绘制干扰线
36          for(int i=0;i<=lineSize;i++){
37              drowLine(g);
38          }
39          //绘制随机字符
40          String randomString = "";
41          for(int i=1;i<=stringNum;i++){
42              randomString=drowString(g,randomString,i);
43          }
44          session.removeAttribute(RANDOMCODEKEY);
45          session.setAttribute(RANDOMCODEKEY, randomString);
46          g.dispose();
47          try {
48              //将内存中的图片通过流动形式输出到客户端
49              ImageIO.write(image, "JPEG", response.getOutputStream());
50          } catch (Exception e) {
51              e.printStackTrace();
52          }
53
54      }
55      /*
56       * 获得字体
```

```
57      */
58     private Font getFont(){
59         return new Font("Fixedsys",Font.CENTER_BASELINE,20);
60     }
61     /*
62      * 获得颜色
63      */
64     private Color getRandColor(int fc,int bc){
65         if(fc > 255)
66             fc = 255;
67         if(bc > 255)
68             bc = 255;
69         int r = fc + random.nextInt(bc-fc-16);
70         int g = fc + random.nextInt(bc-fc-14);
71         int b = fc + random.nextInt(bc-fc-18);
72         return new Color(r,g,b);
73     }
74 
75     /*
76      * 绘制字符串
77      */
78     private String drowString(Graphics g,String randomString,int i){
79         g.setFont(getFont());
80         g.setColor(new
Color(random.nextInt(101),random.nextInt(111),random.nextInt(121)));
81         String rand =
String.valueOf(getRandomString(random.nextInt(randString.length())));
82         randomString +=rand;
83         g.translate(random.nextInt(3), random.nextInt(3));
84         g.drawString(rand, 13*i, 16);
85         return randomString;
86     }
87     /*
88      * 绘制干扰线
89      */
90     private void drowLine(Graphics g){
91         int x = random.nextInt(width);
92         int y = random.nextInt(height);
93         int xl = random.nextInt(13);
94         int yl = random.nextInt(15);
```

```
95          g.drawLine(x, y, x+xl, y+yl);
96      }
97      /*
98       * 获取随机的字符
99       */
100     public String getRandomString(int num){
101         return String.valueOf(randString.charAt(num));
102     }
103 }
104
```