图片验证码

1. 工具类

|  |
| --- |
| package com.sunshine.util;  import java.awt.Color;  import java.awt.Font;  import java.awt.Graphics;  import java.awt.image.BufferedImage;  import java.io.IOException;  import java.io.OutputStream;  import java.util.Random;  import javax.imageio.ImageIO;  import javax.servlet.http.HttpServletResponse;  /\*\*  \*  \* @name DefaultManager  \* @description 图片验证码  \* @author xupf@sunshinepaper.com.cn  \* @version 1.0  \* @date 2017年3月15日  \*/  public class DefaultManager {    private final static int CAPTCHA\_WIDTH = 90;  private final static int CAPTCHA\_HEIGHT = 32;  private final static int CAPTCHA\_FONT\_HEIGHT = 28;  private final static int LINE\_COUNT = 20;  private final static int LINE\_DY = 12;  private static final int MAX\_RGB = 255;  private static final int CAPTCHA\_CODE\_COUNT = 4;  private static final int CAPTCHA\_CODE\_X = 19;// 15  private static final int CAPTCHA\_CODE\_Y = 26;    private static final char[] CAPTCHA\_CODES = "ABCDEFGHJKLMNPQRSTUVWXYZ23456789".toCharArray();    /\*\*  \* 生成captchaCode.  \*  \* @return 随机生成的验证码  \*/  public static String generateCaptchaCode() {  // randomCode用于保存随机产生的验证码，以便用户登录后进行验证。  StringBuilder randomCode = new StringBuilder();  // 创建一个随机数生成器类  Random random = new Random();  // 随机产生codeCount数字的验证码。  for (int i = 0; i < CAPTCHA\_CODE\_COUNT; i++) {  // 得到随机产生的验证码数字。  String code = String.valueOf(CAPTCHA\_CODES[random.nextInt(CAPTCHA\_CODES.length)]);  // 将产生的四个随机数组合在一起。  randomCode.append(code);  }  return randomCode.toString();  }    public static String generateCaptcha(String captchaCode, OutputStream os) throws IOException{  // 定义图像buffer  BufferedImage buffImg = new BufferedImage(CAPTCHA\_WIDTH, CAPTCHA\_HEIGHT, BufferedImage.TYPE\_INT\_RGB);  Graphics gd = buffImg.getGraphics();  // 创建一个随机数生成器类  Random random = new Random();  // 将图像填充为白色  gd.setColor(Color.WHITE);  gd.fillRect(0, 0, CAPTCHA\_WIDTH, CAPTCHA\_HEIGHT);  // 创建字体，字体的大小应该根据图片的高度来定。  Font font = new Font("Fixedsys", Font.BOLD, CAPTCHA\_FONT\_HEIGHT);  // 设置字体。  gd.setFont(font);  // 画边框。  // gd.setColor(Color.BLACK);  gd.drawRect(0, 0, CAPTCHA\_WIDTH - 1, CAPTCHA\_HEIGHT - 1);  // 随机产生40条干扰线，使图象中的认证码不易被其它程序探测到。  gd.setColor(Color.BLACK);  for (int i = 0; i < LINE\_COUNT; i++) {  int x = random.nextInt(CAPTCHA\_WIDTH);  int y = random.nextInt(CAPTCHA\_HEIGHT);  int xl = random.nextInt(LINE\_DY);  int yl = random.nextInt(LINE\_DY);  gd.drawLine(x, y, x + xl, y + yl);  }  // randomCode用于保存随机产生的验证码，以便用户登录后进行验证。  // StringBuffer randomCode = new StringBuffer();  int red = 0, green = 0, blue = 0;  // 随机产生codeCount数字的验证码。  int sw = Math.floorDiv(CAPTCHA\_WIDTH, captchaCode.length());  for (int i = 0; i < captchaCode.length(); i++) {  // 得到随机产生的验证码数字。  String code = String.valueOf(captchaCode.charAt(i));  // 产生随机的颜色分量来构造颜色值，这样输出的每位数字的颜色值都将不同。  red = random.nextInt(MAX\_RGB);  green = random.nextInt(MAX\_RGB);  blue = random.nextInt(MAX\_RGB);  // 用随机产生的颜色将验证码绘制到图像中。  gd.setColor(new Color(red, green, blue));  gd.drawString(code, i \* sw, CAPTCHA\_CODE\_Y);  }  // 将四位数字的验证码保存到Session中。  // redisTemplate.opsForValue().set(getCategory() + ":" + captchaKey, captchaCode, getExpire(), TimeUnit.SECONDS);  // 将图像输出到输出流中。  try (OutputStream ignored = os) {  ImageIO.write(buffImg, "jpeg", os);  }  return captchaCode;  }    public static void main(String[] args) throws IOException {  DefaultManager dm = new DefaultManager();  String string = dm.generateCaptchaCode();  System.out.println(string);  }  } |

1. controller层编写

|  |
| --- |
| @RequestMapping(value = "/restapi/public/userApi/getCaptcha", method = RequestMethod.***GET***, produces = "application/javascript;charset=UTF-8")  **public** **void** getCaptcha(HttpServletResponse resp/\*,  @RequestParam String keyString\*/) **throws** JsonProcessingException {  // if (StringUtils.isNotBlank(keyString)) {  // 禁止图像缓存。  resp.setHeader("Pragma", "no-cache");  resp.setHeader("Cache-Control", "no-cache");  resp.setDateHeader("Expires", 0);  resp.setContentType("image/jpeg");  **try** {  String string = DefaultManager.*generateCaptchaCode*();  DefaultManager.*generateCaptcha*(string, resp.getOutputStream());  } **catch** (IOException e) {  **if** (log.isErrorEnabled()) {  log.error("生成验证码失败", e);  }  }  // }  } |

1. Jsp页面编写

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>图片验证码</title>  <script type=*"text/javascript"* src=*"js/jquery-1.9.0.min.js"*></script>  </head>  <body>  <div class=*"form-group col-lg-6"*>  <label for=*"id"* class=*"col-sm-4 control-label"*>  验证码:  </label>  <div class=*"col-sm-8"*>  <input type=*"text"* id=*"code"* name=*"code"* class=*"form-control"* style="width:*250px*;"/>  <img id=*"imgObj"* alt=*"验证码"* src=*"restapi/public/userApi/getCaptcha"* onclick="changeImg()"/>  <a href=*"#"* onclick="changeImg()">换一张</a>  </div>  </div>    <script type=*"text/javascript"*>  // 刷新图片  **function** changeImg() {  **var** imgSrc = $("#imgObj");  **var** src = imgSrc.attr("src");  alert(src);  imgSrc.attr("src", changeUrl(src));  }  //为了使每次生成图片不一致，即不让浏览器读缓存，所以需要加上时间戳  **function** changeUrl(url) {  /\* var timestamp = (new Date()).valueOf();  var index = url.indexOf("?",url);  if (index > 0) {  url = url.substring(0, url.indexOf(url, "?"));  }  if ((url.indexOf("&") >= 0)) {  url = url + "×tamp=" + timestamp;  } else {  url = url + "?timestamp=" + timestamp;  } \*/  **return** url+"?"+**new** Date().getTime();  }  </script>  </body>  </html> |