1. 引入相关依赖

<dependency>  
 <groupId>com.swetake.util</groupId>  
 <artifactId>util</artifactId>  
 <version>1.0.0</version>  
 <scope>system</scope>  
 <systemPath>${project.basedir}/web/lib/Qrcode\_A.jar</systemPath>  
</dependency>  
<dependency>  
 <groupId>com.swetake.util</groupId>  
 <artifactId>sourceforge</artifactId>  
 <version>1.0.0</version>  
 <scope>system</scope>  
 <systemPath>${project.basedir}/web/lib/qrcode\_B.jar</systemPath>  
</dependency>

1. 引入factory
2. public class QRCodeFactory {  
     
    private static Logger *logger* = LoggerFactory.*getLogger*(QRCodeFactory.class);  
     
    private static final int *PIXOFF* = 2; //偏移量  
     
    */\*\*  
    \* 生成二维码，通过{****@link*** *BufferedImage}返回  
    \** ***@param*** *content 二维码内容  
    \** ***@param*** *qrcodeErrorCorrect N表示数字, A表示a-Z, B表示其他字符  
    \** ***@param*** *qrcodeEncodeMode 纠错等级L(7%)、M(15%)、Q(25%)、H(30%)  
    \** ***@param*** *qrcodeVersion 二维码版本1-40, 版本越高二维码越大  
    \** ***@param*** *encoding 编码格式  
    \** ***@return*** *{****@link*** *BufferedImage}  
    \*/* public static BufferedImage getQRCode(String content, char qrcodeErrorCorrect, char qrcodeEncodeMode, int qrcodeVersion, String encoding) {  
    Qrcode qrcode = new Qrcode();  
    qrcode.setQrcodeEncodeMode(qrcodeEncodeMode); //N表示数字,A表示a-Z, B表示其他字符  
    qrcode.setQrcodeErrorCorrect(qrcodeErrorCorrect); //纠错等级  
    qrcode.setQrcodeVersion(qrcodeVersion); //二维码版本  
     
    int width = 67 + 12 \* (qrcodeVersion - 1);  
    int height = width;  
     
    BufferedImage bufferedImage = new BufferedImage(width, height, BufferedImage.*TYPE\_INT\_RGB*);  
     
    Graphics2D gs = bufferedImage.createGraphics();  
     
    gs.setBackground(Color.*WHITE*);  
    gs.setColor(Color.*BLACK*);  
    gs.clearRect(0, 0, width, height);  
     
    byte[] d = new byte[0];  
    try {  
    d = content.getBytes(encoding);  
    } catch (UnsupportedEncodingException e) {  
    *logger*.error("转换成字节数组出错", e);  
    }  
    if (d.length > 0 && d.length < 120) {  
    boolean[][] s = qrcode.calQrcode(d);  
     
    for (int i = 0; i < s.length; i++) {  
    for (int j = 0; j < s.length; j++) {  
    if (s[j][i]) {  
    gs.fillRect(j \* 3 + *PIXOFF*, i \* 3 + *PIXOFF*, 3, 3);  
    }  
    }  
    }  
    }  
    gs.dispose();  
    bufferedImage.flush();  
    return bufferedImage;  
    }  
     
    */\*\*  
    \* 获取二维码信息，通过String  
    \** ***@param*** *image 图片的{****@link*** *BufferedImage}流  
    \** ***@param*** *encoding 编码格式  
    \** ***@return*** *{****@link*** *String}  
    \*/* public static String decodeQRCode(BufferedImage image, String encoding) {  
    QRCodeDecoder qrCodeDecoder = new QRCodeDecoder();  
    String result = null;  
    try {  
    result = new String(qrCodeDecoder.decode(new QRCodeImageImpl(image)), encoding);  
    } catch (UnsupportedEncodingException e) {  
    *logger*.error("转换成字节数组出错", e);  
    }  
    return result;  
    }  
     
   }

3.使用