package com.keepcodeclean.qrcode;

import com.google.zxing.BarcodeFormat;

import com.google.zxing.EncodeHintType;

import com.google.zxing.client.j2se.MatrixToImageWriter;

import com.google.zxing.common.BitMatrix;

import com.google.zxing.qrcode.QRCodeWriter;

import javax.imageio.ImageIO;

import java.awt.image.BufferedImage;

import java.io.ByteArrayOutputStream;

import java.util.Base64;

import java.util.EnumMap;

import java.util.Map;

import java.util.logging.Logger;

public class QrCodeUtil {

private static final Logger LOG = Logger.getLogger(QrCodeUtil.class.getName());

private static final String BASE64\_PREFIX = "data:image/png;base64,";

private static final int DEFAULT\_QR\_CODE\_WIDTH = 200;

private static final int DEFAULT\_QR\_CODE\_HEIGHT = 200;

private static final int QR\_CODE\_WHITESPACE\_MARGIN = 2;

private static final String DEFAULT\_IMAGE\_FORMAT = "png";

private static final String UTF\_8\_CHARSET = "UTF-8";

public static BufferedImage toQrCode(final String input,

final int width,

final int height) {

final QRCodeWriter barcodeWriter = new QRCodeWriter();

try {

final Map<EncodeHintType, Object> hints = new EnumMap<>(EncodeHintType.class);

hints.put(EncodeHintType.CHARACTER\_SET, UTF\_8\_CHARSET);

hints.put(EncodeHintType.MARGIN, QR\_CODE\_WHITESPACE\_MARGIN);

final BitMatrix bitMatrix = barcodeWriter.encode(input, BarcodeFormat.QR\_CODE, width, height, hints);

return MatrixToImageWriter.toBufferedImage(bitMatrix);

} catch (Exception e) {

LOG.severe("Could not generate a QR code.");

throw new RuntimeException(e);

}

}

public static String toBase64QrCode(final String input,

final int width,

final int height) {

try {

final BufferedImage bufferedImage = toQrCode(input, width, height);

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

ImageIO.write(bufferedImage, DEFAULT\_IMAGE\_FORMAT, outputStream);

return BASE64\_PREFIX + new String(Base64.getEncoder().encode(outputStream.toByteArray()));

} catch (Exception e) {

LOG.severe("Could not generate a QR code.");

throw new RuntimeException(e);

}

}

public static String toBase64QrCode(final String input) {

return toBase64QrCode(input, DEFAULT\_QR\_CODE\_WIDTH, DEFAULT\_QR\_CODE\_HEIGHT);

}

public static BufferedImage toQrCode(final String input) {

return toQrCode(input, DEFAULT\_QR\_CODE\_WIDTH, DEFAULT\_QR\_CODE\_HEIGHT);

}

}