解决rt.jar下sun.misc.BASE64Encoder的依赖

原创

CZ__ 于 2019-03-21 18:24:39 发布 8690 收藏 7
分类专栏： Maven 文章标签： Maven rt.jar BASE64Encoder

本文链接：https://blog.csdn.net/ththcc/article/details/88722325

1. 描述

当使用maven进行打war包时，可能会出现`cannot find symbol`或`sun.misc.BASE64Encoder找不到jar包`。

2. 分析

原因在于BASE64Encoder/BASE64Decoder类在sun.misc包下，是sun公司的内部方法，后期有删除的潜在可能，建议使用`apache commons.codec下的Base64`替代或者jdk自带的`java.util.Base64.Decoder和java.util.Base64.Encoder`的JDK公共API。

java.util.Base64.Decoder的官网API：https://docs.oracle.com/javase/9/docs/api/java/util/Base64.Decoder.html
java.util.Base64.Encoder的官网API：https://docs.oracle.com/javase/9/docs/api/java/util/Base64.Encoder.html

3. 解决方法

方法一、利用JDK替代`sun.misc.BASE64Encoder`
替代前：

```
package util;

import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import javax.imageio.ImageIO;
import org.jbarcode.JBarcode;
import org.jbarcode.encode.Code128Encoder;
import org.jbarcode.paint.BaseLineTextPainter;
import org.jbarcode.paint.EAN13TextPainter;
import org.jbarcode.paint.WidthCodedPainter;

import sun.misc.BASE64Encoder;
```

```
public class BarcodeUtil {

 /**
  * 128条形码
  *
  * @param strBarCode
  *            条形码：0-100位
  * @param dimension
  *            商品条形码：尺寸
  * @param barheight
  *            商品条形码：高度
  * @return 图片(Base64编码)
  */
 public static String generateBarCode128(String strBarCode, String dimension, String barheight) {

  try {
   ByteArrayOutputStream outputStream = null;
   BufferedImage bi = null;
   JBarcode productBarcode = new JBarcode(Code128Encoder.getInstance(), WidthCodedPainter.getInstance(),
     EAN13TextPainter.getInstance());

   // 尺寸，面积，大小 密集程度
   productBarcode.setXDimension(Double.valueOf(dimension).doubleValue());
   // 高度 10.0 = 1cm 默认1.5cm
   productBarcode.setBarHeight(Double.valueOf(barheight).doubleValue());
   // 宽度
   productBarcode.setWideRatio(Double.valueOf(30).doubleValue());
   // 是否显示字体
   productBarcode.setShowText(true);
   // 显示字体样式
   productBarcode.setTextPainter(BaseLineTextPainter.getInstance());

   // 生成二维码
   bi = productBarcode.createBarcode(strBarCode);

   outputStream = new ByteArrayOutputStream();
   ImageIO.write(bi, "jpg", outputStream);

   BASE64Encoder encoder = new BASE64Encoder();
   return encoder.encode(outputStream.toByteArray());
  } catch (Exception e) {
   e.printStackTrace();
   return "encodeError";
  }
 }
}
```

替代为`java.util.Base64.Encoder`后：

```
package util;

import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import javax.imageio.ImageIO;
import org.jbarcode.JBarcode;
import org.jbarcode.encode.Code128Encoder;
import org.jbarcode.paint.BaseLineTextPainter;
```

```
import org.jbarcode.paint.EAN13TextPainter;
import org.jbarcode.paint.WidthCodedPainter;

import java.util.Base64;
import java.util.Base64.Decoder;
import java.util.Base64.Encoder;

public class BarcodeUtil {

 /**
  * 128条形码
  *
  * @param strBarCode
  *            条形码：0-100位
  * @param dimension
  *            商品条形码：尺寸
  * @param barheight
  *            商品条形码：高度
  * @return 图片(Base64编码)
  */
 public static String generateBarCode128(String strBarCode, String dimension, String barheight) {

  try {
   ByteArrayOutputStream outputStream = null;
   BufferedImage bi = null;
   JBarcode productBarcode = new JBarcode(Code128Encoder.getInstance(), WidthCodedPainter.getInstance(),
     EAN13TextPainter.getInstance());

   // 尺寸，面积，大小 密集程度
   productBarcode.setXDimension(Double.valueOf(dimension).doubleValue());
   // 高度 10.0 = 1cm 默认1.5cm
   productBarcode.setBarHeight(Double.valueOf(barheight).doubleValue());
   // 宽度
   productBarcode.setWideRatio(Double.valueOf(30).doubleValue());
   // 是否显示字体
   productBarcode.setShowText(true);
   // 显示字体样式
   productBarcode.setTextPainter(BaseLineTextPainter.getInstance());

   // 生成二维码
   bi = productBarcode.createBarcode(strBarCode);

   outputStream = new ByteArrayOutputStream();
   ImageIO.write(bi, "jpg", outputStream);

   Encoder encoder = Base64.getEncoder();
   return encoder.encodeToString(outputStream.toByteArray());
  } catch (Exception e) {
   e.printStackTrace();
   return "encodeError";
  }
 }

 /**
  * BASE64Decoder 解密
  *
  * @param data 要解密的字符串
  * @return 解密后的byte[]
  * @throws Exception
  */
```

```
 public static byte[] decryptBASE64(String data) throws Exception {
  // BASE64Decoder decoder = new BASE64Decoder();
  // byte[] buffer = decoder.decodeBuffer(data);
  // 从JKD 9开始rt.jar包已废除，从JDK 1.8开始使用java.util.Base64.Decoder
  Decoder decoder = Base64.getDecoder();
  byte[] buffer = decoder.decode(data);
  return buffer;
 }
}
```

方法二、 Maven引入`commons-codec.jar`

```
<dependency>
   <groupId>commons-codec</groupId>
   <artifactId>commons-codec</artifactId>
   <version>1.3</version>
</dependency>
```

替代为`org.apache.commons.codec.binary.Base64`后：

```
package util;

import java.awt.image.BufferedImage;
import java.io.ByteArrayOutputStream;
import javax.imageio.ImageIO;
import org.jbarcode.JBarcode;
import org.jbarcode.encode.Code128Encoder;
import org.jbarcode.paint.BaseLineTextPainter;
import org.jbarcode.paint.EAN13TextPainter;
import org.jbarcode.paint.WidthCodedPainter;

import org.apache.commons.codec.binary.*;;

public class BarcodeUtil {

 /**
  * 128条形码
  *
  * @param strBarCode
  *            条形码：0-100位
  * @param dimension
  *            商品条形码：尺寸
  * @param barheight
  *            商品条形码：高度
  * @return 图片(Base64编码)
  */
 public static String generateBarCode128(String strBarCode, String dimension, String barheight) {

  try {
   ByteArrayOutputStream outputStream = null;
   BufferedImage bi = null;
   JBarcode productBarcode = new JBarcode(Code128Encoder.getInstance(), WidthCodedPainter.getInstance(),
     EAN13TextPainter.getInstance());

   // 尺寸，面积，大小 密集程度
   productBarcode.setXDimension(Double.valueOf(dimension).doubleValue());
   // 高度 10.0 = 1cm 默认1.5cm
   productBarcode.setBarHeight(Double.valueOf(barheight).doubleValue());
```

```
   // 宽度
   productBarcode.setWideRatio(Double.valueOf(30).doubleValue());
   // 是否显示字体
   productBarcode.setShowText(true);
   // 显示字体样式
   productBarcode.setTextPainter(BaseLineTextPainter.getInstance());

   // 生成二维码
   bi = productBarcode.createBarcode(strBarCode);

   outputStream = new ByteArrayOutputStream();
   ImageIO.write(bi, "jpg", outputStream);

   return Base64.encodeBase64String(outputStream.toByteArray());
  } catch (Exception e) {
   e.printStackTrace();
   return "encodeError";
  }
 }

 /**
  * Base64.decodeBase64解密
  *
  * @param data 要解密的字符串
  * @return 解密后的byte[]
  * @throws Exception
  */
 public static byte[] decryptBASE64(String data) throws Exception {
  byte[] buffer = Base64.decodeBase64(data);
  return buffer;
 }
}
```

方法三、 Maven引入`rt.jar`

这种方法还是使用sun.misc.BASE64Encoder，不过不推荐使用，因为依赖于JAVA_HOME，如果其他的服务器没有JAVA_HOME，则会编译失败，报`Fatal Error: Unable to find package java.lang in classpath or bootclasspath`

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>2.5</version>
  <configuration>
   <source>${jdk.version}</source>
   <target>${jdk.version}</target>
   <encoding>UTF-8</encoding>
   <compilerArguments>
    <verbose />
    <bootclasspath>${JAVA_HOME}/jre/lib/rt.jar${path.separator}${JAVA_HOME}/jre/lib/jce.jar</bootclasspath>
   </compilerArguments>
  </configuration>
</plugin>
```