java.util.Base64

# Base64类、Base64.Encoder、Base64.Decoder

三个类：一个Base64类和两个内部类，都存在于java.util包中。都是从JDK1.8添加上去的，算是JDK1.8的新特性。

**java.util.Base64**

**java.util.Base64.Encoder**

**java.util.Base64.Decoder**

# Base64

## 简单介绍

public class **Base64** extends Object

Base64可以看做是一个工具类，内部封装了Base64编码器和解码器，并提供了获取编码器和解码器的静态方法。

Since: 1.8

## 功能介绍

This class **consists** exclusively(专门地，唯一地) of **static methods** for obtaining encoders and decoders for the Base64 encoding scheme. The implementation of this class supports the following types of Base64 as specified *in RFC 4648 and RFC 2045*.

### Basic

Uses "**The Base64 Alphabet**" as specified in Table 1 of RFC 4648 and RFC 2045 for encoding and decoding operation. The encoder does not add any line feed (line separator) character. The decoder rejects data that contains characters outside the base64 alphabet.

### URL and Filename safe

Uses the "**URL and Filename safe Base64 Alphabet**" as specified in Table 2 of RFC 4648 for encoding and decoding. The encoder does not add any line feed (line separator) character. The decoder rejects data that contains characters outside the base64 alphabet.

### MIME

Uses the "**The Base64 Alphabet**" as specified in Table 1 of RFC 2045 for encoding and decoding operation. The encoded output must be represented in lines of no more than 76 characters each and uses a carriage return '\r' followed immediately by a linefeed '\n' as the line separator. No line separator is added to the end of the encoded output. All line separators or other characters not found in the base64 alphabet table are ignored in decoding operation.

### 编码器和解码器方法参数不能为null，否则抛出NullPointerException

Unless otherwise noted, passing a **null** argument to a method of this class will cause a NullPointerException to be thrown.

## 两个内部类

### static class Base64.Decoder：解码

This class implements a **decoder** for decoding byte data using the Base64 encoding scheme as specified in RFC 4648 and RFC 2045.

### static class Base64.Encoder：编码

This class implements an encoder for encoding byte data using the Base64 encoding scheme as specified in RFC 4648 and RFC 2045.

## 静态方法

### getDecoder()和getEncoder()

static Base64.Decoder **getDecoder()**

Returns a Base64.Decoder that decodes using the Basic type base64 encoding scheme.

static Base64.Encoder **getEncoder()**

Returns a Base64.Encoder that encodes using the Basic type base64 encoding scheme.

### getMimeDecoder和getMimeEncoder

static Base64.Decoder **getMimeDecoder**()

Returns a Base64.Decoder that decodes using the MIME type base64 decoding scheme.

static Base64.Encoder **getMimeEncoder**()

Returns a Base64.Encoder that encodes using the MIME type base64 encoding scheme.

static Base64.Encoder **getMimeEncoder**(int lineLength, byte[] lineSeparator)

Returns a Base64.Encoder that encodes using the MIME type base64 encoding scheme with specified line length and line separators.

### getUrlDecoder()和getUrlEncoder()

static Base64.Decoder **getUrlDecoder()**

Returns a Base64.Decoder that decodes using the URL and Filename safe type base64 encoding scheme.

static Base64.Encoder **getUrlEncoder**()

Returns a Base64.Encoder that encodes using the URL and Filename safe type base64 encoding scheme.

继承Object的方法：



# Base64.Encoder

## 简单介绍

public static class **Base64.Encoder** extends Object



存在于java.util包中。 Since: 1.8

没有静态方法，通过Base64的静态方法getEncoder()获取编码器对象。

## 功能介绍

### in RFC 4648 and RFC 2045

This class implements **an encoder** for encoding byte data using the **Base64 encoding scheme** as specified in RFC 4648 and RFC 2045.

### 多线程安全

**Instances of Base64.Encoder class are safe for use by multiple concurrent threads.**

### 不支持null参数，否则NullPointerException

Unless otherwise noted, passing **a null argument** to a method of this class will cause a NullPointerException to be thrown.

## 方法

### encoder方法

**byte[]** encode(byte[] src)

Encodes all bytes from the specified byte array into a newly-allocated byte array using the Base64 encoding scheme.

Parameters: src - the byte array to encode

Returns: A newly-allocated byte array containing the resulting encoded bytes.

**int** encode(**byte[] src, byte[] dst)**

Encodes all bytes from the specified byte array using the Base64 encoding scheme, writing the resulting bytes to the given output byte array, starting at offset 0.

It is the responsibility of the invoker of this method to make sure the output byte array dst has enough space for encoding all bytes from the input byte array. No bytes will be written to the output byte array if the output byte array is not big enough.

Parameters:

src - the byte array to encode

dst - the output byte array

Returns:

**The number of bytes written to the output byte array**

Throws: **IllegalArgumentException** - if dst does not have enough space for encoding all input bytes.

**ByteBuffer** encode(**ByteBuffer buffer**)

Encodes all remaining bytes from the specified byte buffer into a newly-allocated ByteBuffer using the Base64 encoding scheme. Upon return, the source buffer's position will be updated to its limit; its limit will not have been changed. The returned output buffer's position will be zero and its limit will be the number of resulting encoded bytes.

Parameters: buffer - the source ByteBuffer to encode

Returns: A newly-allocated byte buffer containing the encoded bytes.

### encodeToString

public String **encodeToString**(byte[] src)

Encodes the specified byte array into a String using the Base64 encoding scheme.

This method first encodes all input bytes into a base64 encoded byte array and then constructs a new String by using the encoded byte array and **the ISO-8859-1 charset.**

In other words, an invocation of this method has exactly the same effect as invoking **new String(encode(src), StandardCharsets.ISO\_8859\_1)**.(等价于执行这句话)

Parameters: src - the byte array to encode

Returns: A String containing the resulting Base64 encoded characters

### wrap(OutputStream os)

OutputStream wrap(OutputStream os)

Wraps an output stream for encoding byte data using the Base64 encoding scheme.

### withoutPadding()

Base64.Encoder withoutPadding()

Returns an encoder instance that encodes equivalently to this one, but without adding any padding character at the end of the encoded byte data.

# Base64.Decoder

## 简单介绍

public static class Base64.Decoder extends Object

和Base64.Encoder完全对应。

Since: 1.8

## 功能介绍

### **in RFC 4648 and RFC 2045**

This class implements a decoder for decoding byte data using the Base64 encoding scheme as specified **in RFC 4648 and RFC 2045**.

**The Base64 padding character '=' is accepted and interpreted as the end of the encoded byte data,** **but is not required**. So if the final unit of the encoded byte data only has two or three Base64 characters (without the corresponding padding character(s) padded), they are decoded as if followed by padding character(s). If there is a padding character present in the final unit, the correct number of padding character(s) must be present, otherwise IllegalArgumentException ( IOException when reading from a Base64 stream) is thrown during decoding.

### 多线程安全

Instances of Base64.Decoder class are safe for use by multiple concurrent threads.

### 参数不能null，否则NullPointerException

Unless otherwise noted, passing a null argument to a method of this class will cause a NullPointerException to be thrown.

## 方法

### decode

**byte[]** **decode(byte[] src)**

Decodes all bytes from the input byte array using the Base64 encoding scheme, writing the results into a newly-allocated output byte array.

**int decode(byte[] src, byte[] dst)**

Decodes all bytes from the input byte array using the Base64 encoding scheme, writing the results into the given output byte array, starting at offset 0.

**ByteBuffer decode(ByteBuffer buffer)**

Decodes all bytes from the input byte buffer using the Base64 encoding scheme, writing the results into a newly-allocated ByteBuffer.

### 把字符串解析为字节数组

byte[] decode(String src)

Decodes a Base64 encoded String into a newly-allocated byte array using the Base64 encoding scheme.

### wrap(InputStream is)

InputStream wrap(InputStream is)

Returns an input stream for decoding Base64 encoded byte stream.

# 使用示例

获取标准的Base64编解码器：

**Base64.Encoder base64Encoder = Base64.getEncoder();**

**Base64.Decoder base64Decoder = Base64.getDecoder();**

String str = "中国是一个伟大的国家！山东省菏泽曹县是一个大县城。中国是一个伟大的国家！山东省菏泽曹县是一个大县城。";

helper(base64Encoder,base64Decoder,str);

获取UrlBase64编解码器：

**base64Encoder = Base64.getUrlEncoder();**

**base64Decoder = Base64.getUrlDecoder();**

// helper(base64Encoder,base64Decoder,str);//仅仅只是将+变成-，将/变成\_

获取MimeBase64编解码器

**base64Encoder = Base64.getMimeEncoder();**

**base64Decoder = Base64.getMimeDecoder();**

helper(base64Encoder,base64Decoder,str);

**public** **static** **void** helper(Base64.Encoder base64Encoder,Base64.Decoder base64Decoder,String str){

String result = base64Encoder.encodeToString(str.getBytes());

System.***out***.println("Base64编码前结果："+str);

System.***out***.println("Base64编码后结果："+result);

String original = **new** String(base64Decoder.decode(result));

System.***out***.println("Base64解码后结果："+original);

}

