




sun.misc.BASE64Encoder详解

原创

我叫吴桂鑫  于 2017-11-24 18:47:05 发布  3318  收藏 3

分类专栏: [java](#) 文章标签: [编码](#) [base64](#) [java](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/baidu_36583119/article/details/78627369

版权



[java](#) 专栏收录该内容

42 篇文章 0 订阅

订阅专栏

(一)、BASE64编码规则及JAVA中的使用

##1、编码规则:

Base64编码要求把3个8位字节 (3*8=24) 转化为4个6位的字节 (4*6=24), 之后在6位的前面补两个0, 形成8位一个字节的 形式。

例如字符串“张3”:

11010101 11000101 00110011

用十进制表示即为: 53 34 20 51

这个并不是最终的结果, 还需要根据Base64的编码表查询出转换后值。

下面就是**BASE64**编码表:

Value	Encoding	Value	Encoding	Value	Encoding	Value	Encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v	(pad)	=
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

1. 以上一共是64个编码，这也是Base64名称的由来。
2. 编码的编号（1-64）对应的是得出的新字节的十进制值。因此，上例中字符串“张3”经过编码后就成了字符串“1iUz”了。
3. 若要对只有两个字节的字符串进行编码，如“张”，则在其编码结果的后面加“=”，即“1iU=”。
4. 若要对只有一个字节的字符串进行编码，如“3”，则在其编码结果的后面加“=”，即“z=”。

##2、编码和解码

在JAVA中要实现Base64的编码和解码是非常容易的，因为JDK中已经有提供有现成的类：

编码：

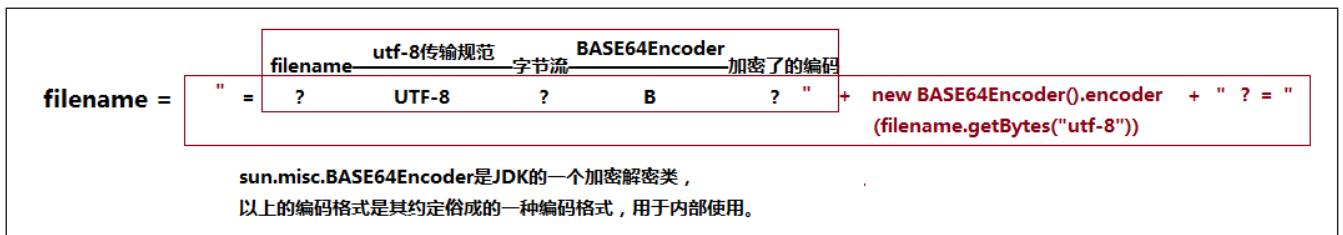
```
String src ="BASE64编码测试";
sun.misc.BASE64Encoder en = new sun.misc.BASE64Encoder();
String encodeStr = en.encode(src.getBytes());
```

解码：

```
sun.misc.BASE64Decoder dec = newsun.misc.BASE64Decoder();
byte[] data = dec.decodeBuffer(encodeStr);
```

[更多知识点参考请点击这里](#)

```
filename = "? UTF-8?B?" + new BASE64Encoder().encoder(filename.getBytes("utf-8")) + "?=";
```



###BASE64Encoder 加密和解密

```
public class B64Demo {
    public static String getBASE64(String s) {
        if (s == null)
            return null;
        return (new sun.misc.BASE64Encoder()).encode(s.getBytes());
    }
    // 将 BASE64 编码的字符串 s 进行解码
    public static String getFromBASE64(String s) {
        if (s == null)
            return null;
        sun.misc.BASE64Decoder decoder = new sun.misc.BASE64Decoder();
        try {
            byte[] b = decoder.decodeBuffer(s);
            return new String(b);
        } catch (Exception e) {
            return null;
        }
    }
    // 将 BASE64 编码的字符串 s 进行加密，即对字符串进行三次的BASE64编码
    public static String encryption(Object obj){
        return B64Demo.getBASE64(B64Demo.getBASE64(B64Demo.getBASE64((String)obj)));
    }
    // 将 BASE64 编码的字符串 s 进行解密，即对字符串进行三次的BASE64解码
    public static String decryption(String 3b64){
        return B64Demo.getFromBASE64(B64Demo.getFromBASE64(B64Demo.getFromBASE64(3b64)));
    }
    public static void main(String[] args) {
        String a = encryption("100000.89".toString());
        System.out.println(a);//加密
        System.out.println(decryption(a));//解密
    }
}
```