2016-06-26 23:59:47 1540浏览 1评论

手记 \ Base64编码原理解析与Java实现

Base64编码原理解析与Java实现

小字号 ▼

我的购物车

购物车里空空如也

快去这里选购你中意的课程

实战课程

职业路径

已加入0门课程

去购物车

:

?

0

2篇手 我的订单中心

作者的热门手记 Java常用字符串反转的五种方案 975浏览 8推荐 0评论

小瑕疵,所以文章格式怪怪的。虽然CSDN的格式也略怪,但好歹没有漏掉一些符号,建议 看原文吧。另,晚些时候再尝试慕课网的手记功能好了。 一、前言

声明:本文首发在CSDN的个人博客(点此直达首发),慕课网的手记功能目前还存在一些

分享到 🚱 💣 💂

目前还在找工作,工作日时投投简历面面试,这周末难免就闲来无事了,那就只好看看慕课逛

逛CSDN了,正巧看到一个关于Base64的课程《Java实现Base64加密》,点进去看看,完了发 觉完全不是我想的那回事儿,人给的实现方式还不唯一,给了3个API,但是没有实现原理。我这 个愣头青没别的优点,就是喜欢死钻牛角尖,于是抱着试试的心态,查了官网RFC2045(下载完 整PDF),了解了下相关情况,这才有了本文。

二、Base64是什么

这种高度概括的事情我相信Base64的百度百科比我靠谱多了,如下: Base64是网络上最常见的用于传输8Bit字节代码的编码方式之一,大家可以查看RFC2045~

RFC2049, 上面有MIME的详细规范。Base64编码可用于在HTTP环境下传递较长的标识信息。 例如,在JavaPersistence系统Hibernate中,就采用了Base64来将一个较长的唯一标识符(一 般为128-bit的UUID)编码为一个字符串,用作HTTP表单和HTTPGETURL中的参数。在其他应 用程序中,也常常需要把二进制数据编码为适合放在URL(包括隐藏表单域)中的形式。此时, 采用Base64编码具有不可读性,即所编码的数据不会被人用肉眼所直接看到。 标准的Base64并不适合直接放在URL里传输,因为URL编码器会把标准Base64中的"/"和"+"字 符变为形如"%XX"的形式,而这些"%"号在存入数据库时还需要再进行转换,因为ANSISQL中已

将"%"号用作通配符。 为解决此问题,可采用一种用于URL的改进Base64编码,它在末尾填充'='号,并将标准Base64 中的"+"和"/"分别改成了"-"和"", 这样就免去了在URL编解码和数据库存储时所要作的转换, 避免

了编码信息长度在此过程中的增加,并统一了数据库、表单等处对象标识符的格式。 另有一种用于正则表达式的改进Base64变种,它将"+"和"/"改成了"!"和"-",因为"+","*"以及前面 在IRCu中用到的"["和"]"在正则表达式中都可能具有特殊含义。

此外还有一些变种,它们将"+/"改为"-"或"."(用作编程语言中的标识符名称)或".-"(用于XML中 的Nmtoken)甚至":"(用于XML中的Name)。

Base64要求把每三个8Bit的字节转换为四个6Bit的字节(38=46=24),然后把6Bit再添两位高 位0,组成四个8Bit的字节,也就是说,转换后的字符串理论上将要比原来的长1/3。

三、规则与原理

原理(这回觉得百度百科不靠谱,引用的是Base64的维基百科)

在MIME格式的电子邮件中,base64可以用来将binary的字节序列数据编码成ASCII字符序列构

成的文本。使用时,在传输编码方式中指定base64。使用的字符包括大小写字母各26个,加上10 个数字,和加号"+",斜杠"/",一共64个字符,等号"="用来作为后缀用途。 完整的base64定义可见RFC1421和RFC2045。编码后的数据比原始数据略长,为原来的4/3。

在电子邮件中,根据RFC822规定,每76个字符,还需要加上一个回车换行。可以估算编码后数 据长度大约为原长的135.1% (算式1(4/3)((76+1)/76)约等于1.351) 。 转换的时候,将三个byte的数据,先后放入一个24bit的缓冲区中,先来的byte占高位。数据不足 3byte的话,于缓冲器中剩下的bit用0补足。然后,每次取出6(因为26=64)个bit,按照其值选

择ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/中的字符作为

当原数据长度不是3的整数倍时,如果最后剩下一个输入数据,在编码结果后加2个"=";如果最后 剩下两个输入数据,编码结果后加1个"=";如果没有剩下任何数据,就什么都不要加,这样才可 以保证数据还原的正确性。

位一个字节的形式。

上面大段的文字概括来讲,可以总结为以下3点:

编码后的输出。不断进行,直到全部输入数据转换完成。

如果剩下的字符不足3个字节,则用0填充,输出字符使用'=',因此编码后输出的文本末尾可能会 出现1或2个'=';

①把3个8位字节(38=24)转化为4个6位的字节(46=24),之后在6位的前面补两个0,形成8

②每76个字节数据后加一个换行符; ③若数据长度除以3余1,则在编码结束时加2个"=",若数据长度除以3余2,则在编码结束时加1

1"="

规则 一个是ASCII编码表,如下:

ASCII 字符代码表 一

ASCII 打印字符

低四位		0000				1				2		3		4		5		6		7				
																						十进制		ctrl
		000	0	0	BLANK NULL	^@	NUL	空	16	•	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	,	112
001	1	1	0	^ A	SOH	头标开始	17	•	^Q	DC1	设备控制 1	33	!	49	1	65	Α	81	Q	97	а	113	q	
010	2	2	•	^в	STX	正文开始	18	1	^R	DC2	设备控制 2	34	"	50	2	66	В	82	R	98	b	114	r	
011	3	3	٧	^c	ETX	正文结束	19	!!	^s	DC3	设备控制 3	35	#	51	3	67	С	83	S	99	С	115	s	
100	4	4	•	^D	EOT	传输结束	20	1	^ T	DC4	设备控制 4	36	\$	52	4	68	D	84	Т	100	d	116	t	
0101	5	5	*	^ E	ENQ	查询	21	9	οu	NAK	反确认	37	%	53	5	69	E	85	U	101	е	117	u	
110	6	6	٨	^F	ACK	确认	22	•	^ V	SYN	同步空闲	38	&	54	6	70	F	86	٧	102	f	118	v	
0111	7	7	•	^G	BEL	震铃	23	1	^ W	ETB	传输块结束	39	•	55	7	71	G	87	w	103	g	119	w	
000	8	8		^ H	BS	退格	24	1	^x	CAN	取消	40	(56	8	72	Н	88	Х	104	h	120	х	
1001	9	9	0	^I	TAB	水平制表符	25	1	^ Y	EM	媒体结束	41)	57	9	73	1	89	Y	105	i	121	У	
1010	A	10	0	^J	LF	换行/新行	26	\rightarrow	^z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z	
011	В	11	ď	^ K	VT	竖直制表符	27	←	1 ^	ESC	转意	43	+	59	;	75	K	91	[107	k	123	{	
1100	С	12	Q	^L	FF	換页/新页	28	L	^1	FS	文件分隔符	44	,	60	<	76	L	92	1	108	1	124	1	
101	D	13	₽	^H	CR	回车	29	+	^]	GS	組分隔符	45	-	61	=	77	M	93	1	109	m	125	}	
110	E	14	.7	îĦ	50	移出	30	A	^6	RS	记录分隔符	46		62	>	78	Ν	94	^	110	n	126	~	
1111	F	15	a	^0	SI	移入	31	•	^-	US	单元分隔符	47	1	63	?	79	0	95	_	111	0	127	Δ	Bac spac
													注	:: 表中	中的本	SCII等	符可	以用:	ALT -	+ */\	上盘鲈	上的数	字键"	输)

Value Encoding Value Encoding Value Encoding 0 A 17 R 1 B 18 S

2 C

3 D

下:

原数据

ASCII编码

原数据

19 T 36 k 20 U 37 1 21 V

Table 1: The Base64 Alphabet

34 i

35 j

51 z

53 1

E 2	21 V	38 m	55 3						
F 2	22 W	39 n	56 4						
G 2	23 X	40 o	57 5						
H 2	24 Y	41 p	58 6						
I 2	25 Z		59 7						
J 2		43 r	60 8						
K 2	27 b	44 s	61 9						
L 2	28 c	45 t	62 +						
M 2	29 d	46 u	63 /						
N 3	30 e	47 v							
0 3	31 f	48 W	(pad) =						
P 3	32 g	49 x	17 70						
Q 3	33 h	50 y							
实例									
我们还且来丢下面过3个例之比较直现(例之下裁)									
13川灯を不住「凹凸」「削丁心秋且別(削丁下秋)。									
	F G H I J K L M N O P Q	F 22 W G 23 X H 24 Y I 25 Z J 26 a K 27 b L 28 c M 29 d N 30 e O 31 f P 32 g Q 33 h	F 22 W 39 n G 23 X 40 o H 24 Y 41 p I 25 Z 42 q J 26 a 43 r K 27 b 44 s L 28 c 45 t M 29 d 46 u N 30 e 47 v O 31 f 48 w P 32 g 49 x Q 33 h 50 y						

manus colui a	· · ·			-	-			•

例子2、	假设我们的明	l文为"test" (数据长度为	14 , 4%3=1) ,则	其编码计算	方式如下:	
原数据	t	e			t			

101

25

例子1、假设我们的明文为"Base64"(数据长度为6,正好是3的倍数),则其编码计算方式如

索引 21 51 29 Base64编码

例子3、假设我们的明文为"JiaMi"(数据长度为5,5%3=2),则其编码计算方式如下:

二进制位 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0

ASCII编码	74	1	05	97	77	1	05	
二进制位	0 1 0 0 1 0	1 0 0 1 1 0	1 0 0 1	0 1 1 0 0 0 0 1	0 1 0 0 1 1	0 1 0 1 1 0	1 0 0 1 0	0
索引	18	38	37	33	19	22	36	
ase64编码	S	n	1	h	T	¥	k	=

六、代码

import java.util.List;

package com.dikio.base64;

思路什么的,我觉得就不必要写了,代码里注释得很充分了。

import java.util.ArrayList; public class Base64 { private static String base64Code = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghi public static void main(String[] args) { System.out.println(encode("Base64")); System.out.println(decode("QmFzZTY0")); public static String encode(String srcData) { if(srcData == null srcData.length() == 0) { return srcData; char[] chArr = srcData.toCharArray(); String asciiBin = null; StringBuilder asciiBin_all = new StringBuilder(); for(int i= 0; i< chArr.length; i++) { //将字符转换成ASCII编码再转换成对应二进位 asciiBin = Integer.toBinaryString((int)chArr[i]); //给不足8位的在高位补0直到补足8位 while(asciiBin.length()< 8) { 相关标签: JAVA

相关阅读



· Java中String与byte[]的转换	
·eclipse修改编码格式的方式	

总算找到一个不自相矛盾的代码

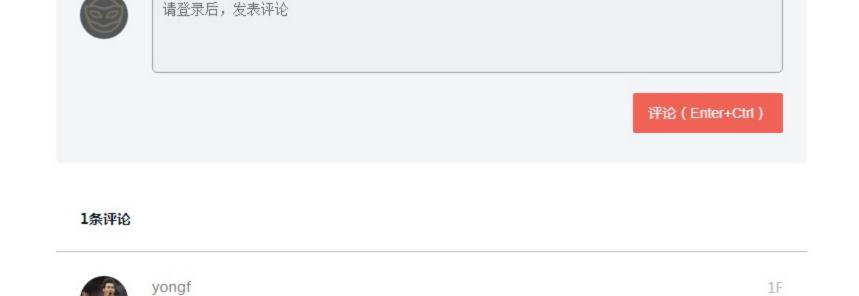
Copyright © 2017 imooc.com All Rights Reserved | 京ICP备 13046642号-2

回复

0

企业合作

·Java中的编码学习笔记



2017-07-17 18:50:34