<https://mp.weixin.qq.com/s/Y0-nku0-QvGNNYkoiFTVLg>

一文彻底看懂Base64原理（并使用java实现）

Base64编码解码算是网络安全领域的很小而且很简单的一个知识点了，虽然简单但是应用场景却极其广泛，可以这样说，开发者要是不懂Base64，基本上可以告别程序猿生涯了。这篇文章从原理入手，并给出java的Base64实现。方便你掌握Base64。

### 一、Base64由来

很早之前，电子邮件刚刚问世，那时候消息的传递都是英文，后来中国开通了互联网之后，对邮件的使用量也大量增加，这时候电子邮件就有了中文的需求。但是中文在传输的时候不能被有效地处理，这时候Base就出来了，Base64通过对这些中文进行编码，转化为服务器和网关能够识别的数据。这时候就能够使用电子邮件有效地传输了。

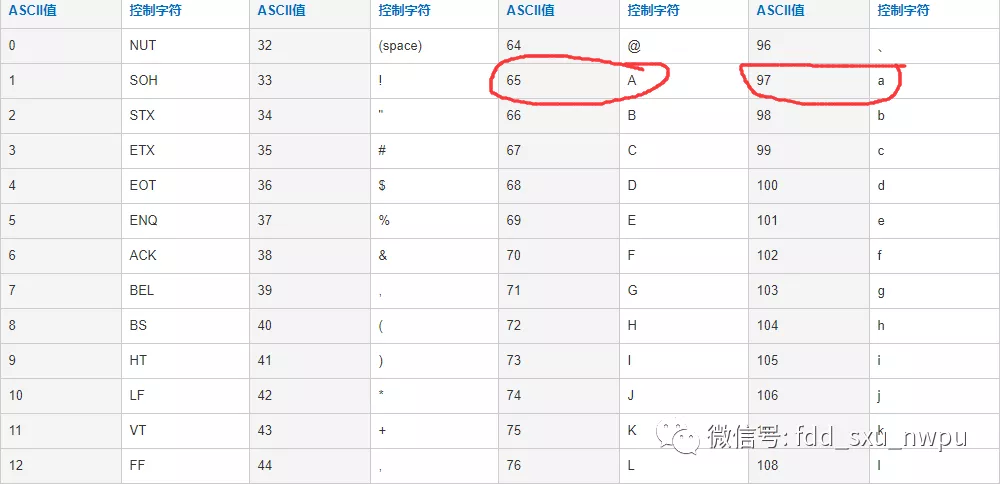
上面的这个例子不是说Base64专门为电子邮件而生的，从其诞生之初，就开始在各大领域有了广泛的应用。比如说网络上传递图片，我们可以Base64先对图片进行处理，然后就可以有效的传输了。

OK，我们大致知道其用途，然后我们就好好的分析一下他的原理，到底是如何对这些数据进行编码的。

### 二、Base64原理

#### 1、Base64编码

Base64的原理超级简单，相信我们都知道ASCII 编码，从A-Z、a-z、0-9和一些其他的特殊字符，这些字符都有唯一的一个数字来表示。比如说a是97，A是65。我们来截取一部分图看一下：



同理Base64也有这样一套编码。范围是”A-Z“、”a-z“、”0-9“、”+“、”/“一共64个字符。我们给出一个表格来看一下，这个比ASCII编码要简单多了，只有64个。

|  |
| --- |
|  |
| **索引** | **对应字符** | **索引** | **对应字符** | **索引** | **对应字符** | **索引** | **对应字符** |
| 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 |  |  |
| 14 | O | 31 | f | 48 | w |  |  |
| 15 | P | 32 | g | 49 | x |  |  |
| 16 | Q | 33 | h | 50 | y |  |  |

由于索引是从0开始，所以最后的索引是63。在编码的时候Base64就是通过上面的进行转换编码的。下面我们就来看看Base64编码的原理。

#### 2、基本原理

比如说有一封邮件，我们想要对其使用Base64进行编码。怎么办呢？基本步骤如下：

（1）对邮件的数据进行切分，每三个字节一组，一共24个bit。

（2）对切分后的数据重组，24个bit重组为4组，每组6个bit。

（3）对重组后的数据处理，每组最前面添加两个“0”，构成每组8个bit。此时一共32个bit。

（4）根据Base64编码表，获取相应的编码值。

此时一封完整的邮件，被切分重组处理之后就变成了Base64编码了。基本原理其实很简单。不过你不理解也没关系，我们直接上个实例来解释一下。

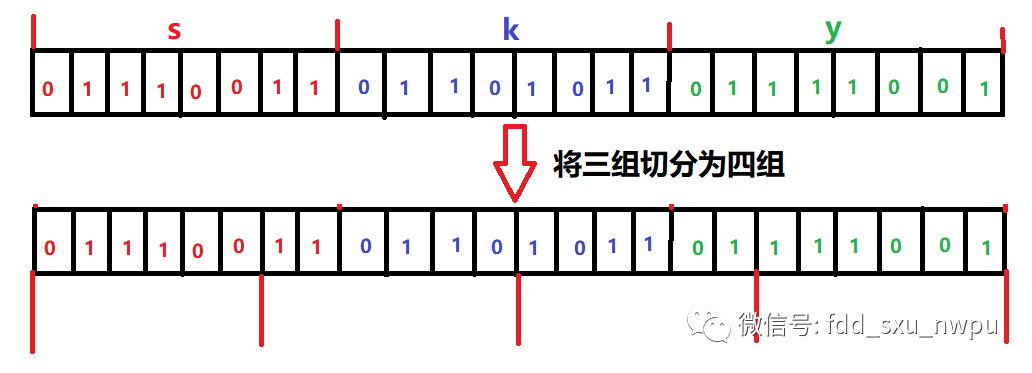
#### 3、实例验证

比如说电子邮件里面出现了三个字母sky。我们要对这个三个字符使用Base64进行编码。

（1）对邮件的数据进行切分，每三个字节一组，一共24个bit

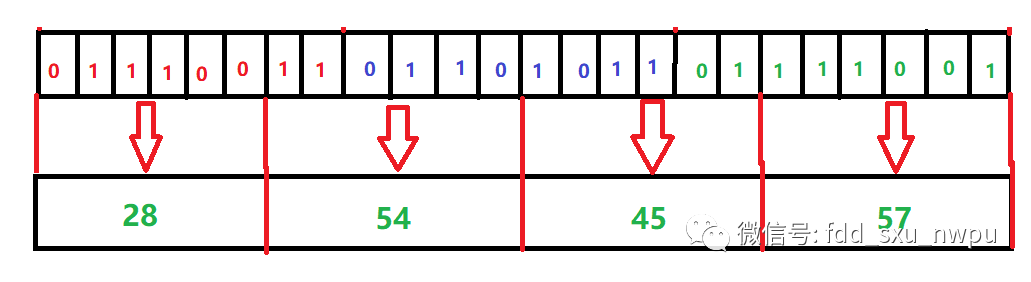
| **数据** | **s** | **k** | **y** |
| --- | --- | --- | --- |
| ASCII编码 | 115 | 107 | 121 |
| 二进制 | 01110011 | 01101011 | 01111001 |

（2）对切分后的数据重组，24个bit重组为4组，每组6个bit

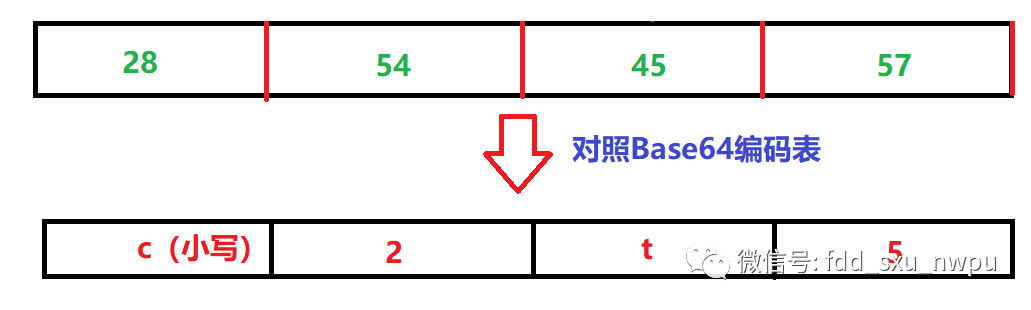


（3）对重组后的数据处理，每组最前面添加两个“0”，构成每组8个bit。由于在最前面添加的0，所以对数值不构成影响。

（4）根据Base64编码表，获取相应的编码值



（5）完成编码的转换



到这我们基本上就是实现了Base64编码机制从sky到c2t5的转换。

**有些地方需要我们去注意一下：**

**（1）在第三步中，最前面添加了两个0，所以最终编码之后要比之前多出三分之一的大小。**

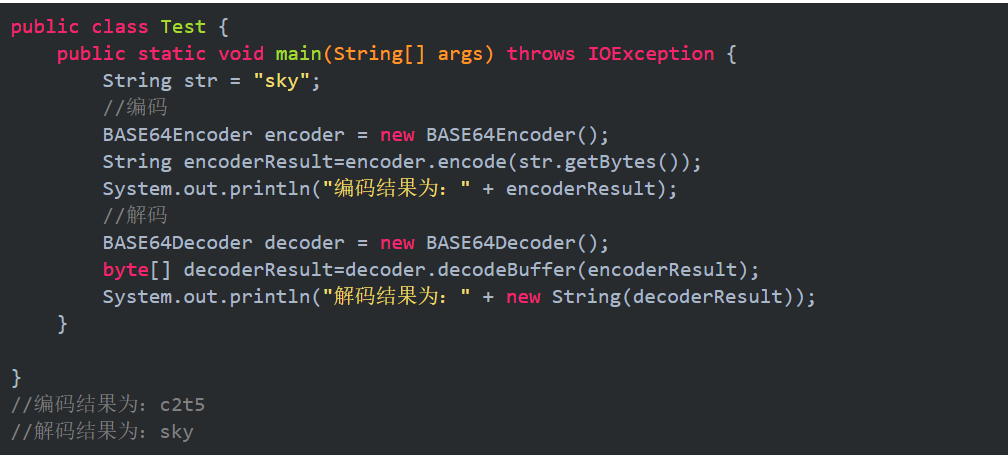
**（2）上面的例子中，我们使用的是ASCII编码，但是如果我们使用UTF-8，对应Base64编码的结果是不一样的。**

**（3）Base64只是进行了编码，方便数据的传输而已。这可不是加密。**

原理也搞清楚了，现在我们就实现一下。

## 三、代码实现

你可以自己去实现一个编码解码的完整过程，但是java已经为我们封装好了，我们直接只用别人造好的轮子多好。不管是自己写还是使用别人的，原理搞清楚就OK了。



Base64算法的应用场景有E-Mail、密钥、证书文件等等。