

Java解析网络数据流的三种特殊方法 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/273/2021\\_2022\\_Java\\_E8\\_A7\\_A3\\_E6\\_9E\\_90\\_c104\\_273313.htm](https://www.100test.com/kao_ti2020/273/2021_2022_Java_E8_A7_A3_E6_9E_90_c104_273313.htm) 介绍用Java解析网络数据的三种特殊方法，希望对正在寻求这方面问题答案的读者有所帮助。 UTF8转换成GB2312 当我们在基于HTTP协议的JSP或Servlet的应用中获取数据或发送请求时，JVM会把输送的数据编码成UTF8格式。如果我们直接从HTTP流中提取中文数据，提取的结果为“????”（可能更多问号），为转换成我们能够理解的中文字符，我们需要把UTF8转换成GB2312，借助ISO-8859-1标准编码能够轻易的实现，下面的代码实现了这一功能：以下是引用片段：

```
byte [] b;String utf8_value.  
utf8_value = request.getParameter("NAME");//从HTTP流中  
取"NAME"的UTF8数据b = utf8_value.getBytes("8859_1"). //中间  
用ISO-8859-1过渡String name = new String(b, "GB2312"). //转换  
成GB2312字符在知道流长度的情况下将输入流转换成字节数组  
Java中的输入流抽象类InputStream有int read(byte[] b, int off,  
int len)方法，参数中byte[] b是用来存放从InputStream中读取  
的数据,int off指定数组b的偏移地址，也就是数组b的起始下标  
，int len指定需要读取的长度，方法返回实际读取的字节数。  
刚学Java的朋友可能要说：先定义一个与流长度等长的字节数  
组，调用read方法，指定起始下标为0，指定读取长度与数组  
长度等长，不是一下子可以读出来了吗？说的没错，笔者曾  
经也试着这样读取数据，但后来发现在读取网络数据时很不  
安全，我们想想在网络上获取数据可能并没那么流畅，数据  
流的传送可能会断断续续，所以并不能保证一次就能读取全
```

部数据，特别是在读取大容量数据时更是如此，所以我们在读取数据时检测实际读到的长度，如果没有读完已知长度的数据就应该再次读取，以此循环检测，直到实际读取的长度累加与已知的长度相等，下面的代码实现了这一功能：

```
ServletInputStream inStream =
request.getInputStream(). //取HTTP请求流
int size =
request.getContentLength(). //取HTTP请求流长度
byte[] buffer =
new byte[size]. //用于缓存每次读取的数据
byte[] in_b = new
byte[size]. //用于存放结果的数组
int count = 0.
int rbyte = 0.
while
(count < size)
rbyte = inStream.read(buffer). //每次实际读取长度存
于rbyte中
for(int i=0; i < count; i++)
in_b[i] = buffer[i].}
count = rbyte.}
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问  
[www.100test.com](http://www.100test.com)