

Java网络编程 - JavaSocket编程（四）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Java_E7_BD_91_E7_BB_9C_c104_144416.htm 重复和并发服务器 这个应用程序被当作一个重复的服务器.因为它只有在处理完一个进程以后才会接受另一个连接.更多的复杂服务器是并发的。它为每一个请求分配一个线程，而不是来一个处理一个。所以看起来它在同时处理多人请求.所有的商业的服务器都是并发的服务器。Java数据报类不像面向连接的类，数据报的客户端和服务器的类在表面上是一样的。下面的程序建立了一个客户和服务器的数据报sockets：
`DatagramSocket serverSocket = new DatagramSocket(4545);`
`DatagramSocket clientSocket = new DatagramSocket();`
服务器用参数4545来指定端口号，由于客户端将要呼叫服务器，客户端可以利用可利用的端口.如果省略第二个参数，程序会让操作系统分配一个可用的端口。客户端可以请求一个指定的端口，但是如果其它的应用程序已经绑定到这个端口之上，请求将会失败.如果你的意图不是作为一个服务器，最好不要指定端口。由于流不能由交谈得到，那么我怎么如何与一个数据报Socket进行对话。答案在于数据报类。接收数据报 DatagramPacket类是用来通过DatagramSocket类接收和发送数据的类。packet类包括了连接信息和数据.就如前面所说的一样，数据报是自身独立的传输单元。
。DatagramPacket类压缩了这些单元。下面的程序表示了用一个数据报socket来接收数据：
`DatagramPacket packet = new DatagramPacket(new byte[512], 512);`
`clientSocket.receive(packet);`
`clientSocket.receive(packet);`
packet的构建器需要知道将得到的

数据放在哪儿。一个512字节的缓存被建立并且作为构建器的第二个参数。第二个构建器参数是缓存的大小。就像ServerSocket类的accept()方法一样，receive()方法在数据可用之前将会阻塞。发送数据报发送数据报是非常地简单地，所有需要的只是一个地址。地址是由InetAddress类来建立的。这个类没有公共的构建器，但是它有几个static的方法，可以用来建立这个类的实例。下面的列表列出了建立InetAddress类的实例的方法：

```
Public InetAddress Creation  
Methods  
InetAddress getByName(String host).  
InetAddress[] getAllByName(String host).  
InetAddress getLocalHost().
```

得到本地主机的地址是非常地有用的，只有前面两个方法是用来发送数据包的。getByName()和getAllByName()需要目的主机的地址。第一个方法仅仅只是返回第一个符合条件的东西。第二个方法是必须的，因为一台计算机可能有多个地址。在这种情况下，这台计算机被称为multi-homed。所有的建立的方法都被标记为static。它们必须像下面这样得到调用：

```
InetAddress addr1 = InetAddress.getByName("merlin").  
InetAddress addr2[] = InetAddress.getAllByName("merlin").  
InetAddress addr3 = InetAddress.getLocalHost().
```

重复和并发服务器 所有的这些调用都可以掷出一个UnknownHostException违例。如果一台计算机没有连接上DNS服务器，或者主机的确没有找到，这个违例就会被掷出。如果一台计算机没有一个激活的TCP/IP配置，getLocalHost()也为失败并掷出一个违例。一旦一个地址被确定了，数据报就可以被送出了。下面的程序传输了一个字符串给目的socket：

```
String toSend = "This is the data to send!";  
byte[] sendbuf = new byte[ toSend.length() ];
```

toSend.getBytes(0, toSend.length(), sendbuf, 0). DatagramPacket
sendPacket = new DatagramPacket(sendbuf, sendbuf.length, addr,
port). clientSocket.send(sendPacket). 首先，字符串必须被转换
成一个字节数组。然后，一个新的DatagramPacket实例必须被
建立.注意构建器的最后两个参数.因为要发送一个包，所以地
址和端口必须被给定。一个applet可能可以知道它的服务器的
地址，但是服务器如何知道它的客户机的地址呢。当任何一
个包被收到后，返回的地址和端口会被解压出来，并通
过getAddress()和getPort()方法得到。这就是一个服务器如何回
应一个客户端的包： DatagramPacket sendPacket = new
DatagramPacket(sendbuf, sendbuf.length, recvPacket.getAddress(),
recvPacket.getPort()). serverSocket.send(sendPacket). 不像面向
连接的操作，数据报服务器服务器其实比数据报客户端更简
单：一个数据报服务器的基本步骤：1.在一个指定的端口上
建立一个数据报socket. 2.用receive方法等待进来的包. 3.用特定
的协议来回应收到的包. 4.回到第二步或继续第二步. 5.关闭数
据报socket. 列表9.3演示了—人简单的数据报回应服务器.它将
回应它收到的包. 列表9.3.一个简单的数据报回应服务器

```
import java.io.*; import java.net.*; public class  
SimpleDatagramServer { public static void main(String[] args) {  
DatagramSocket socket = null; DatagramPacket recvPacket,  
sendPacket; try { socket = new DatagramSocket(4545); while (socket  
!= null) { recvPacket= new DatagramPacket(new byte[512], 512).  
socket.receive(recvPacket); sendPacket = new  
DatagramPacket(recvPacket.getData(),  
recvPacket.getLength(),recvPacket.getAddress(),
```

```
recvPacket.getPort() ). socket.send( sendPacket ). } } catch  
(SocketException se) { System.out.println("Error in  
SimpleDatagramServer: " se). } catch (IOException ioe) {  
System.out.println("Error in SimpleDatagramServer: " ioe). } } }
```

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