

和网络编程有关的基本API位于java.net包中。`import java.net.*;`以java.net.Socket类代表客户端连接，以java.net.ServerSocket类代表服务器端连接。

### “请求-响应”模式：

1. Socket类：发送TCP消息。代表网络连接。
2. ServerSocket类：创建服务器。

套接字是一种进程间的数据交换机制。单个套接字是一个端点，而一对套接字则构成一个双向通信信道，使非关联进程可以在本地或通过网络进行数据交换。一旦建立套接字连接，数据即可在相同或不同的系统中双向或单向发送，直到其中一个端点关闭连接。套接字与主机地址和端口地址相关联。主机地址就是IP地址。端口地址是通信端口。

在客户端和服务端中，分别创建独立的Socket，并通过Socket的属性，将两个Socket进行连接，这样，客户端和服务端通过套接字所建立的连接使用输入输出流进行通信。

TCP/IP套接字是最可靠的双向流协议，使用TCP/IP可以发送任意数量的数据。

实际上，套接字只是计算机上已编号的端口。如果发送方和接收方计算机确定好端口，他们就可以通信了。

### TCP/IP通信连接的简单过程：

位于A计算机上的TCP/IP软件向B计算机发送包含端口号的消息，B计算机的TCP/IP软件接收该消息，并进行检查，查看是否有它知道的程序正在该端口上接收消息。如果有，他就将该消息交给这个程序。（测试一个端口一次通信）

### 通过Socket的编程顺序：

1. 创建服务器ServerSocket，在创建时，定义ServerSocket的监听端口(在这个端口接收客户端发来的消息)。
2. ServerSocket调用accept()方法，使之处于阻塞状态。  
和IO部分的read方法一样，都是一个阻塞方法，也就是当无连接时，该方法将阻塞程序的执行，直到连接到达时才执行该行代码。另外获得的连接会在服务器端的该端口注册，这样以后就可以通过在服务器端的注册信息直接通信，而注册以后服务器端的端口就被释放出来，又可以继续接受其它的连接了。
3. 创建客户端Socket，并设置服务器的IP及端口。
4. 客户端发出连接请求，建立连接。
5. 分别取得服务器和客户端Socket的InputStream和OutputStream。
6. 利用Socket和ServerSocket进行数据传输。
7. 关闭流及Socket。

## 两种模式：多数据交换和支持多个客户端

**服务器端：**一个线程专门发送消息，一个线程专门接收消息。

**客户端：**一个线程专门发送消息，一个线程专门接收消息。  
或是通过多线程一个服务器服务多个客户端。