

通过前面的笔记我们可以知道：一个客户端想要发起一次通信，先决条件就是需要知道运行着服务器端程序的主机的IP地址是多少。然后我们才能够通过这个地址向服务器发送信息。

获取主机地址信息

在Java中我们使用InetAddress类来代表目标网络地址，包括主机名和数字类型的地址信息，并且InetAddress的实例是不可变的，每个实例始终指向一个地址。InetAddress类包含两个子类，分别对应两个IP地址的版本：

Inet4Address

Inet6Address

我们通过前面的笔记可以知道：IP地址实际上是分配给主机与网络之间的连接，而不是主机本身，NetworkInterface类提供了访问主机所有接口的信息的功能。下面我们通过一个简单的示例程序来学习如何获取网络主机的地址信息：

```
1.  import java.net.*;
2.  import java.util.Enumeration;
3.
4.  public class InetAddressExample {
5.
6.      /**
7.       * @param args
8.       */
9.      public static void main(String[] args) {
10.         // TODO Auto-generated method stub
11.         try {
12.             // 获取主机网络接口列表
13.             Enumeration<NetworkInterface> interfaceList = NetworkInterfac
14. e
15.                 .getNetworkInterfaces();
16.             // 检测接口列表是否为空,即使主机没有任何其他网络连接,回环接口(loopbac
17. k)也应该是存在的
18.             if (interfaceList == null) {
19.                 System.out.println("--没有发现接口--");
20.             } else {
21.                 while (interfaceList.hasMoreElements()) {
22.                     // 获取并打印每个接口的地址
23.                     NetworkInterface iface = interfaceList.nextElement();
24.
25.                     // 打印接口名称
26.                     System.out.println("Interface" + iface.getName() + ";
27. ");
28.                     // 获取与接口相关联的地址
29.                     Enumeration<InetAddress> addressList = iface
30.                         .getInetAddresses();
31.                     // 是否为空
32.                     if (!addressList.hasMoreElements()) {
33.                         System.out.println("\t(没有这个接口相关的地址)");
34.                     }
35.                 }
36.             }
37.         } catch (Exception e) {
38.             System.out.println("Exception: " + e);
39.         }
40.     }
41. }
```

```

30.         }
31.         // 列表的迭代, 打印出每个地址
32.         while (addressList.hasMoreElements()) {
33.             InetAddress address = addressList.nextElement();
34.
35.             System.out
36.                 .print("\tAddress"
37.                     + ((address instanceof Inet4Address ? "(v4)"
38.                        : address instanceof Inet6Address ? "(v6)"
39.                           : "(?)"));
40.             System.out.println(": " + address.getHostAddress()
41. );
42.         }
43.     } catch (SocketException se) {
44.         System.out.println("获取网络接口错误:" + se.getMessage());
45.     }
46.     // 获取从命令行输入的每个参数所对应的主机名和地址, 迭代列表并打印
47.     for (String host : args) {
48.         try {
49.             System.out.println(host + ":");
50.             InetAddress[] addressList = InetAddress.getAllByName(host
51. );
52.             for (InetAddress address : addressList) {
53.                 System.out.println("\t" + address.getHostAddress() + "/"
54.                     + address.getHostAddress());
55.             }
56.         } catch (UnknownHostException e) {
57.             System.out.println("\t无法找到地址:" + host);
58.         }
59.     }
60. }
61. }

```

查看运行效果:

```
C:\WINDOWS\system32\cmd.exe

D:\Console_Java>javac InetAddressExample.java
D:\Console_Java>java InetAddressExample
InterfaceI0;
  Address(v4):127.0.0.1
Interfaceeth0;
  Address(v4):192.168.1.86

D:\Console_Java>java InetAddressExample www.baidu.com
InterfaceI0;
  Address(v4):127.0.0.1
Interfaceeth0;
  Address(v4):192.168.1.86
www.baidu.com:
  www.baidu.com/119.75.217.56
  www.baidu.com/119.75.218.77

D:\Console_Java>
```

总结:

在Java Socket编程中获取主机地址信息最关键的两个类是:InetAddress和NetworkInterface, 在Socket编程时必须了解这两个类的常用API和并且熟练的使用它们。

参考资料:《TCP/IP Socket in Java》