

利用网络 Socket 和多线程实现一个双向聊天

作者: [Sysecho](#)

原文链接: <https://ld246.com/article/1498788483428>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

接收键盘输入然后向对方发送消息的线程

```
package cn.com.chat;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.Socket;

/**
 *
 * @author Xiaofei
 *
 */
public class SendMsgThread extends Thread{
    /**
     * 用来接收键盘输入然后向对方发送消息的线程
     */
    Socket ss = null;
    BufferedWriter bw = null;
    BufferedReader br = null;
    public SendMsgThread(Socket ss){
        this.ss = ss;
        try {
            bw = new BufferedWriter(new OutputStreamWriter(ss.getOutputStream()));
            br = new BufferedReader(new InputStreamReader(System.in));
        } catch (IOException e) {
            // TODO 自动生成的 catch 块
            e.printStackTrace();
        }
    }
    public void run(){
        try {
            while(true){
                String str;
                str = br.readLine();
                if(str.equals("exit")){
                    System.exit(0);
                }else{
                    bw.write(str+"\n");
                    bw.flush();
                }
            }
        } catch (IOException e) {
            // TODO 自动生成的 catch 块
            e.printStackTrace();
        }finally{
            try {
                bw.close();
                br.close();
            }
        }
    }
}
```

```
        } catch (IOException e) {
            // TODO 自动生成的 catch 块
            e.printStackTrace();
        }
    }
}
}
```

服务端：Server

```
package cn.com.chat;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

public class Server {
    public static void main(String[] args) {
        ServerSocket server = null;
        Socket socket = null;
        BufferedReader br = null;
        try {
            server = new ServerSocket(88);
            socket = server.accept();
            new SendMsgThread(socket).start();
            br = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            while(true){
                String str = br.readLine();
                System.out.println("客户端: \n" + str);
            }
        } catch (IOException e) {
            // TODO 自动生成的 catch 块
            e.printStackTrace();
        } finally {
            try {
                br.close();
                server.close();
            } catch (IOException e) {
                // TODO 自动生成的 catch 块
                e.printStackTrace();
            }
        }
    }
}
```

客户端：Client

```
package cn.com.chat;
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;

public class Client {
    public static void main(String[] args) {
        Socket socket = null;
        BufferedReader br = null;
        try {
            socket = new Socket("192.168.16.111", 88);
            System.out.println("请输入您的消息: ");
            new SendMsgThread(socket).start();
            br = new BufferedReader(new InputStreamReader(
                socket.getInputStream()));
            while (true) {
                String str = br.readLine();
                System.out.println("客户端: \n" + str);
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                br.close();
                socket.close();
            } catch (IOException e) {
                // TODO 自动生成的 catch 块
                e.printStackTrace();
            }
        }
    }
}
```

特别提醒一点：注意启动的顺序，先启动服务端，再启动客户端。现接收的服务器，客户端才能够获得服务端的端口和IP。