

Program

server.java

```
import java.io.*;
import java.util.*;
import java.net.*;

public class Server {

    static Vector<ClientHandler> ar = new Vector<>();

    static int i = 0;

    public static void main(String[] args) throws IOException {
        ServerSocket ss = new ServerSocket(8002);

        Socket s;

        while (true) {
            s = ss.accept();
            System.out.println("New client request received : " + s);

            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new
DataOutputStream(s.getOutputStream());
            System.out.println("Creating a new handler for this
client...");

            ClientHandler mtch = new ClientHandler(s, "client " + i, dis,
dos);

            Thread t = new Thread(mtch);
            System.out.println("Adding this client to active client list");

            ar.add(mtch);
            t.start();
            i++;
        }
    }

    class ClientHandler implements Runnable {
        Scanner scn = new Scanner(System.in);
        private String name;
        final DataInputStream dis;
        final DataOutputStream dos;
        Socket s;
        boolean isloggedin;

        public ClientHandler(Socket s, String name, DataInputStream dis,
DataOutputStream dos) {
            this.dis = dis;
```

```

        this.dos = dos;
        this.name = name;
        this.s = s;
        this.isloggedin = true;
    }

    @Override
    public void run() {

        String received;
        while (true) {
            try {
                received = dis.readUTF();

                System.out.println(received);

                if (received.equals("logout")) {
                    this.isloggedin = false;
                    this.s.close();
                    break;
                }

                StringTokenizer st = new StringTokenizer(received, "#");
                String MsgToSend = st.nextToken();
                String recipient = st.nextToken();

                for (ClientHandler mc : Server.ar) {
                    if (mc.name.equals(recipient) && mc.isloggedin == true)
                    {
                        mc.dos.writeUTF(this.name + " : " + MsgToSend);
                        break;
                    }
                }
            } catch (IOException e) {}
        }
        try {
            this.dis.close();
            this.dos.close();
        } catch (IOException e) {}
    }
}

```

client.java

```

import java.io.*;
import java.net.*;
import java.util.Scanner;

public class Client {
    final static int ServerPort = 8002;

```

```

    public static void main(String args[]) throws UnknownHostException,
IOException {
        Scanner scn = new Scanner(System.in);
        InetAddress ip = InetAddress.getByName("localhost");
        Socket s = new Socket(ip, ServerPort);

        DataInputStream dis = new DataInputStream(s.getInputStream());
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());

        Thread sendMessage = new Thread(new Runnable() {
            @Override
            public void run() {
                while (true) {
                    String msg = scn.nextLine();

                    try {
                        dos.writeUTF(msg);
                    } catch (IOException e) {}
                }
            }
        });

        Thread readMessage = new Thread(new Runnable() {
            @Override
            public void run() {
                while (true) {
                    try {
                        String msg = dis.readUTF();
                        System.out.println(msg);
                    } catch (IOException e) {}
                }
            }
        });
        sendMessage.start();
        readMessage.start();
    }
}

```

Output

Screenshots

```
abhayvashokan@abhayvashokan: ~/Network Programming Lab/6
abhayvashokan@abhayvashokan:~/Network Programming Lab/6$ javac UDPServer.java
[[Abhayvashokan@abhayvashokan:~/Network Programming Lab/6$ java UDPServer
Server socket created
From Client: Hello World
Server socket created
From Client: Exit
Server Exit...
abhayvashokan@abhayvashokan:~/Network Programming Lab/6$
```

```
abhayvashokan@abhayvashokan: ~/Network Programming Lab/6
abhayvashokan@abhayvashokan:~/Network Programming Lab/6$ javac UDPClient.java
abhayvashokan@abhayvashokan:~/Network Programming Lab/6$ java UDPClient
Client socket created
Message to server: Hello World
Message to server: Exit
Message to server:
```

Output

Server

```
Server socket created
From Client: Hello World
Server socket created
From Client: Exit
Server Exit...
```

Client

```
Client socket created
Message to server: Hello World
Message to server: Exit
```

ReadMe

1. Open first terminal
 1. `javac server.java`
 2. `java server`
2. Open second terminal
 1. `javac client.java`
 2. `java client`
3. Communicate between Client and Server using the terminal.
4. To exit type: `Exit` in Client.