

Computer Networks - Exp 6

Kartik Jolapara

60004200107 - B1

Aim

To implement socket communication in java.

Theory

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while another socket reaches out to the other to form a connection. Server forms the listener socket while the client reaches out to the server.

Java Socket programming is used for communication between the applications running on different JRE. Java Socket programming can be connection-oriented or connectionless. Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connectionless socket programming.

The client in socket programming must know two information:

1. IP Address of Server
2. Port number

Socket Class

A socket is simply an endpoint for communications between the machines. The Socket class can be used to create a socket.

ServerSocket Class

The ServerSocket class can be used to create a server socket. This object is used to establish communication with the clients.

User Datagram Protocol(UDP)

DatagramSockets are Java's mechanism for network communication via UDP instead of TCP. Java provides DatagramSocket to communicate over UDP instead of TCP. It is also built on top of IP. DatagramSockets can be used to both send and receive packets over the Internet.

One of the examples where UDP is preferred over TCP is the live coverage of TV channels. In this aspect, we want to transmit as many frames to a live audience as possible without worrying about the loss of one or two frames. TCP being a reliable protocol adds its own overhead while transmission. Another example where UDP is preferred is online multiplayer gaming. In games like counter- strike or call of duty, it is not necessary to relay all the information but the most important ones. It should also be noted that most of the applications in real life use a careful blend of both UDP and TCP; transmitting the critical data over TCP and the rest of the data via UDP.

Code(TCP)

Server

```
import java.io.*;

import java.net.*;

class Server {

    public static void main(String[] args) throws Exception {

        String msg;

        ServerSocket ss = new ServerSocket(80);

        while (true) {
```

```
Socket s1 = ss.accept();

String week[] = { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday" };

int i = (int) (Math.random() * week.length);

msg = week[i];

PrintStream ps = new PrintStream(s1.getOutputStream());

ps.println(msg);

}

}

}
```

Client

```
import java.io.*;

import java.net.*;

class Client {

    public static void main(String[] args) throws Exception {

        Socket cs = new Socket("localhost", 80);

        BufferedReader br = new BufferedReader(new
InputStreamReader(cs.getInputStream()));

        String m = br.readLine();

        System.out.println("Message from server = " +m);
```

```
        cs.close();  
    }  
}
```

Output

<pre>d:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>java Server █</pre>	<pre>D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>java Client Message from server = Sunday D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>java Client Message from server = Saturday D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>java Client Message from server = Friday D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>java Client Message from server = Tuesday D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>java Client Message from server = Saturday D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>java Client Message from server = Thursday D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\Simple Client Server>█</pre>
---	--

Code(UDP)

Server

```
import java.io.IOException;  
  
import java.net.DatagramPacket;  
  
import java.net.DatagramSocket;  
  
import java.net.InetAddress;  
  
import java.net.SocketException;
```

```
public class Server {  
  
    public static void main(String[] args) throws IOException {  
  
        DatagramSocket ds = new DatagramSocket(1234);  
  
        byte[] receive = new byte[65535];  
  
        DatagramPacket DpReceive = null;  
  
        while (true) {  
  
            DpReceive = new DatagramPacket(receive, receive.length);  
  
            ds.receive(DpReceive);  
  
            System.out.println("Client:-" + data(receive));  
  
            if (data(receive).toString().equals("bye")) {  
  
                System.out.println("Client sent bye. EXITING");  
  
                break;  
  
            }  
  
            receive = new byte[65535];  
  
        }  
  
    }  
  
}
```

```
public static StringBuilder data(byte[] a) {  
  
    if (a == null)  
  
        return null;  
  
    StringBuilder ret = new StringBuilder();
```

```
    int i = 0;

    while (a[i] != 0) {

        ret.append((char) a[i]);

        i++;

    }

    return ret;

}

}
```

Client

```
import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.util.Scanner;


public class Client {

    public static void main(String args[]) throws IOException {

        Scanner sc = new Scanner(System.in);

        DatagramSocket ds = new DatagramSocket();

        InetAddress ip = InetAddress.getLocalHost();

        byte buf[] = null;
```

```
while (true) {  
  
    String inp = sc.nextLine();  
  
    buf = inp.getBytes();  
  
    DatagramPacket DpSend = new DatagramPacket(buf, buf.length, ip, 1234);  
  
    ds.send(DpSend);  
  
    if (inp.equals("bye"))  
  
        break;  
  
}  
  
}  
  
}
```

Output

<pre>d:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\UDP Server>java Server Client:-Hi there Client:-how are you Client:-this is me Client:-and am going.. Client:-now.. Client:-bye Client sent bye. EXITING d:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\UDP Server></pre>	<pre>D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\UDP Server>java Client Hi there how are you this is me and am going.. now.. bye D:\DJSCE\Practicals\SEM 4\CN\CN Exp 6 - Socket Programming\UDP Server></pre>
--	---

Conclusion

We have Successfully implemented Socket programming using TCP and UDP protocols in JAVA.