

## Assignment No 2

31/08/2020

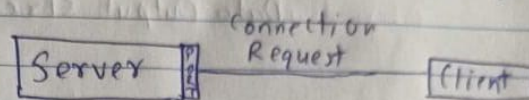
- Title : Socket Programming
- Problem Statement: Enhance the system with the help of socket programming using client-server architecture development chat server.
- Objective : To understand and implement socket programming.
- Outcome : We will be able to :
  - 1) Design and implement socket programming in JAVA.
  - 2) Implement TCP/UDP client-server socket based application.
- SW & HW requirements :

Eclipse IDE, Fedora OS / windows 10, Intel i5 processor, 4 GB RAM.
- Theory :

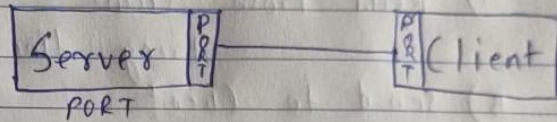
Java creates applications using socket. Programs running on client makes a request to a program running on server, involves networking server provided by transport layer. Transport layer has two protocols (TCP, UDP). These use port to map a particular process on computer. Multiple services can run at same time so we distinguish between them.

A socket is a software end point that establishes a bidirectional common between a server program one or more client programs. The socket associates the server with a specific hardware port on ~~one~~<sup>the</sup> machine where it runs, so a client program wherever in network with a socket associate with same port can communicate with server program.

A) Client making a request



B) Connection Established



java.net package :

It contains key classes and interfaces which simplifies client-server programs.

→ Classes : Content Handler, Datagram,

Connection, Content Handler, InetAddress, Multicast, socket, server socket, URL, URL connection.

→ Interfaces - Content Handler, Factory, File Name HCP, socket.



## TCP/IP Socket Programming

### Server:

- 1) Open server sockets `server socket s = new ServerSocket(port)`
- 2) wait for client request socket `c = s.accept()`;
- 3) Create I/O streams for communication -  
`DataInputStream is = new DataInputStream(c.getInputStream());`  
`DataOutputStream os = new DataOutputStream(c.getOutputStream());`
- 4) Read from client  
`String line = (String) is.readUTF();`
- 5) send to client  
`os.writeBytes("Hello\n");`
- 6) Close socket: `c.close()`

### Client:

- 1) Create socket object  
`Socket c = new Socket(server, port-id);`
- 2) DataInputStream is = new DataInputStream(c.getInputStream());
- 3) DataOutputStream os = new DataOutputStream(c.getOutputStream());
- 4) Send data to server  
`os.writeBytes("Hello");`
- 5) Close socket: `client.close();`

## UDP Socket Programming :

TCP guarantees delivery of packets and preserves the order on destination. Sometimes these features are not required and since they do not come without performance costs, it would be better to use higher transport protocol.

Datagram packets are used to implement a connectionless packet delivery service supported by UDP. Each message is transferred from source machine to destination based on information contained within packet. This means each packet needs to have destination address and each packet might be routed differently.

## Serializable Interface

Serialization is a mechanism of converting the state of an object into a byte stream. Deserialization is the reverse process of the byte stream which is used to recreate the actual Java object in memory.

The byte stream created is platform independent so the object serialized on one platform can be used by deserializing on different. To make the Java object serializable we implement `java.io.Serializable` interface.



## Test Cases

I/P Choose Movie Genre : comedy

O/P Movie Recommended : The Hangover

I/P Choose Movie Genre : adventure

O/P Movie Recommended : Mulan

I/P Choose Movie Genre : ~~Get Out~~ horror

O/P Movie Recommended : Get out

I/P Choose Movie Genre : comedy

O/P movie Recommended : Step Brothers

I/P Choose Movie Genre : adventure

O/P Movie Recommended : Dune

I/P Choose Movie Genre : horror

O/P movie Recommended : World War Z

Server.java

```
import java.io.*;
```

```
import java.net.*;
```

```
import java.util.*;
```

```
class Server {
```

```
    // public static void CheckOutServer() throws Exception {
```

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

```
            Random rand = new Random();
```

```
            ArrayList<String> comedy= new ArrayList<String>();
```

```
            ArrayList<String> adventure= new ArrayList<String>();
```

```
            ArrayList<String> horror= new ArrayList<String>();
```

```
            horror.add("A quiet place");
```

```
            horror.add("Get out");
```

```
            horror.add("Alien");
```

```
            horror.add("World War Z");
```

```
            adventure.add("Mulan");
```

```
            adventure.add("The Old Guard");
```

```
            adventure.add("Dune");
```

```
            adventure.add("Wonder Woman 1984");
```

```
            comedy.add("Step Brothers");
```

```
            comedy.add("White Chicks");
```

```
            comedy.add("The Hot chick");
```

```
            comedy.add("The Hangover");
```

```
            System.out.println("S : Server started ");
```

```
            // Create server Socket
```

```
            ServerSocket ss = new ServerSocket(4449);
```

```
            System.out.println("S : Server waiting.. ");
```

```
            // connect it to client socket
```

```

Socket s = ss.accept();

System.out.println("S : Connection established ");


// to send data to the client

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


// to read data coming from the client

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


// to read data from the keyboard

BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));


//DataInputStream din = new DataInputStream(s.getInputStream());


String str, str1 = "";


str = br.readLine();

//str = din.readUTF();

System.out.println("S : Client data: "+ str);


if(str.equals("comedy"))
    { str1 = comedy.get(rand.nextInt(comedy.size()));}
else if (str.equals("adventure"))
    { str1 = adventure.get(rand.nextInt(adventure.size()));}
else if (str.equals("horror"))
    { str1 = horror.get(rand.nextInt(horror.size()));}
else{
    str1 = "Enter valid choice please\n";
}

ps.println(str1);

```

```

        System.out.println("S : Data sent ");
        // close connection
        ps.close();
        br.close();
        kb.close();
        ss.close();
        s.close();

        // terminate application
        System.exit(0);
    }
}

```

Client.java

```

import java.io.*;
import java.net.*;

class Client {

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

        System.out.println("C : In Client ");
        // Create client socket
        Socket s = new Socket("localhost", 4449);

        // to send data to the server
        DataOutputStream dos = new DataOutputStream(s.getOutputStream());

        // to read data coming from the server
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
    }
}

```



```
// to read data from the keyboard
BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));

String str, str1;

System.out.println("Choose Movie Genre(Enter lowercase
name):\n1.Comedy\t2.Adventure\t3.Horror");

str = kb.readLine();
System.out.println("Genre Chosen");
dos.writeBytes(str + "\n");

// receive from the server
str1 = br.readLine();

System.out.println("C : Movie recommended: " + str1);

// close connection
dos.close();
br.close();
kb.close();
s.close();
}
}
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Durvesh\Downloads> java Server.java
S : Server started
S : Server waiting..
S : Connection established
S : Client data: comedy
S : Data sent
PS C:\Users\Durvesh\Downloads> java Server.java
S : Server started
S : Server waiting..
S : Connection established
S : Client data: adventure
S : Data sent
PS C:\Users\Durvesh\Downloads> java Server.java
S : Server started
S : Server waiting..
S : Connection established
S : Client data: horror
S : Data sent
PS C:\Users\Durvesh\Downloads> _
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.450]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Durvesh\Downloads> java Client.java
C : In Client
Choose Movie Genre(Enter lowercase name):
1.Comedy      2.Adventure   3.Horror
comedy
Genre Chosen
C : Movie recommended: The Hangover

C:\Users\Durvesh\Downloads> java Client.java
C : In Client
Choose Movie Genre(Enter lowercase name):
1.Comedy      2.Adventure   3.Horror
adventure
Genre Chosen
C : Movie recommended: Mulan

C:\Users\Durvesh\Downloads> java Client.java
C : In Client
Choose Movie Genre(Enter lowercase name):
1.Comedy      2.Adventure   3.Horror
horror
Genre Chosen
C : Movie recommended: Get out

C:\Users\Durvesh\Downloads> _
```