

Introduction To Network Programming - ITS1140

Module Outline

Research doc

Chat Application

Network Terminologies

IP Address

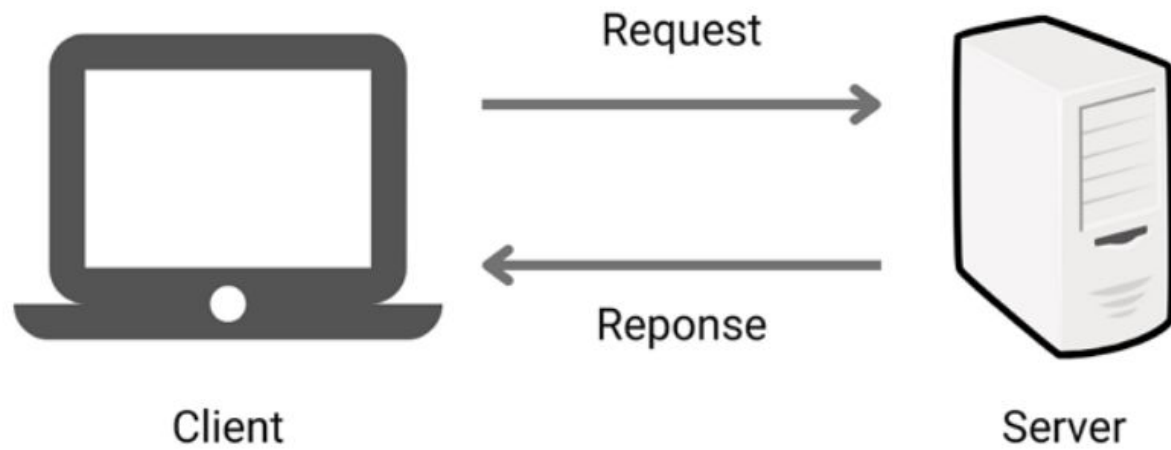
Protocall

Port Number

TCP -

UDP

Client-Server Architecture



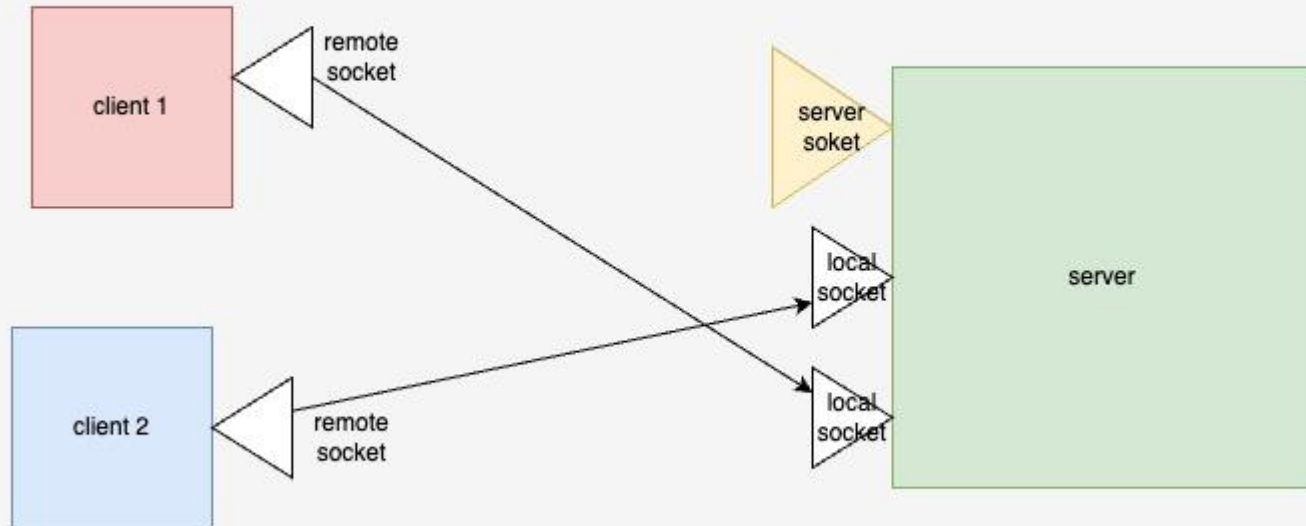
Introduction To Java Socket Programming

- **What Is Socket Programming in Java ?**

Socket Programming is a way of connecting two nodes on a network to communicate with each other.

One Socket (node) listens on a specific port at an IP, While other socket reaches out to the other in order to form a connection

Socket Programming



What is a Socket ?

- A Socket in Java is one **endpoint** of a two way communication link between two programs running on the network.
- A socket is bound to a **port number** so that the TCP layer can identify the application that data is destined to be sent to.

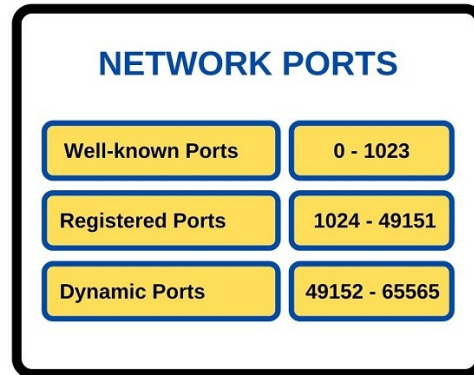


What is an Endpoint ?

- An endpoint is a combination of an **IP address** and a port number.
- The package in the Java platform provides a class, `Socket` that implements one side of a two-way connection between your Java program and another program on the network.

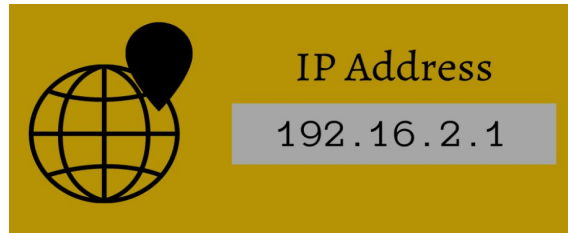
What is a Port Number ?

- A port number is a way to identify a specific process to which an internet or other network message is to be forwarded when it arrives at a server.



What is an IP Address ?

- An IP address is a unique address with a numerical value that identifies a device on the internet or a local network.
- IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network.



What is **Client Side** Programming ?

- In the case of client-side programming, the client will first wait for the server to start.
- Once the server is up and running, it will send the requests to the server. After that, the client will wait for the response from the server.
- This is the whole logic of client and server communication.

Client Side Programming

- **Steps to initiate a Client Request**
 1. **Establishing a connection.**
 2. **Communication.**
 3. **Closing the connection.**

How to Establish a connection ?

- The very first step is to establish a **socket connection**. A socket connection implies that the two machines have information about each other's network location (**IP Address**) and **TCP port**.
- You can create a Socket by using the below statement :

```
Socket socket = new Socket("127.0.0.1", 5000)
```

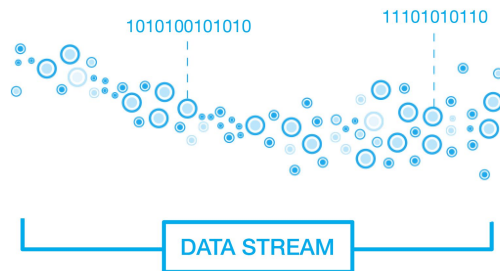
- Represents the Ip Address Of the server (First argument).
- Represents the TCP Port (Second Argument).

How to Communicate with a socket ?

- In order to communicate over a socket connection, **streams** are used for both **input** and **output** the data.
- After establishing a connection and sending the requests, you need to close the connection.
- **getOutputStream()** method is used to send the output through the socket.

What is a data Stream ?

- A stream is a sequence of data. In Java, a stream is composed of bytes. It's called a stream because it is like a stream of water that continues to flow.



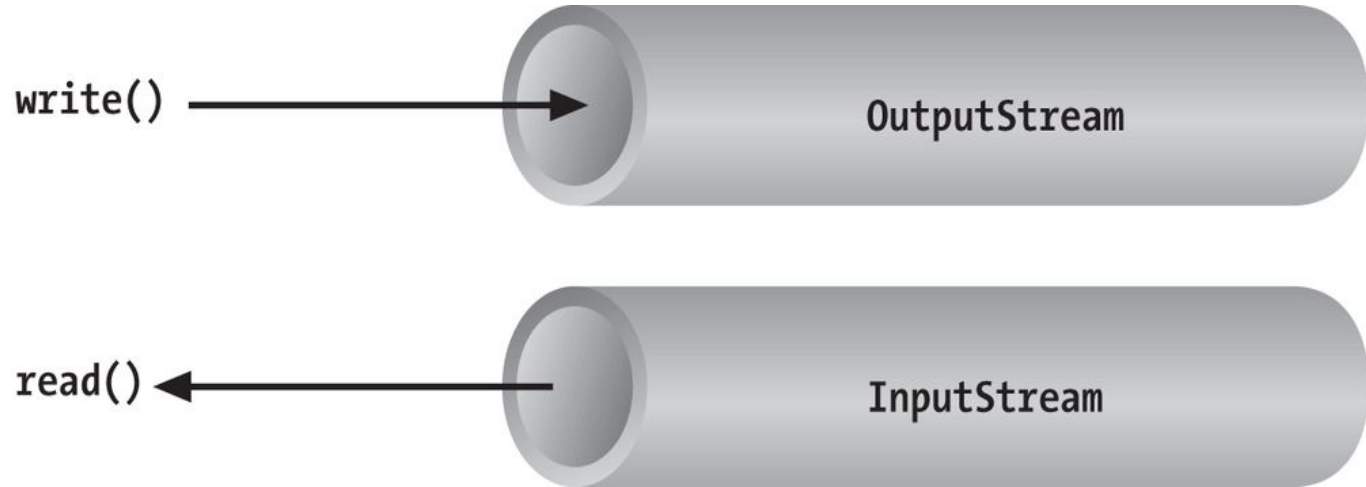
Streams used In Socket Programming.

1. InputStream

- Java application uses an input stream to read data from a source; it may be a file, an array, peripheral device or socket.

1. OutputStream

- Java application uses an output stream to write data to a destination; it may be a file, an array, peripheral device or socket.



Closing The Connection

- The socket connection is closed explicitly once the message to the server is sent.
- It is important to close the connection by closing the socket as well as input/output streams once everything is done.
- Java program to implement socket connection at client side is to be coded.....

End Of Client Side Programming...

What is **Server Side** Programming ?

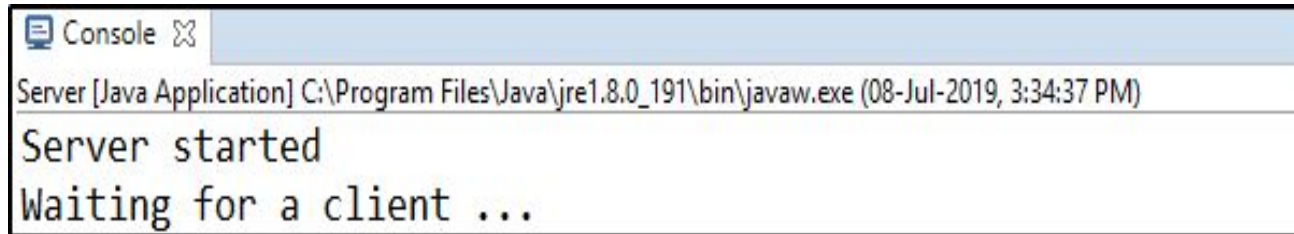
- Basically, the server will instantiate its object and wait for the client request.
- Once the client sends the request, the server will communicate back with the response.
- In order to code the server-side application, you need two sockets and they are as follows :
 - A `ServerSocket` which waits for the client requests (when a client makes a new `Socket()`)
 - A plain old socket for communication with the client.

The Final OutPut ..

- After configuring both client and server end, you can execute the server side program first.
- After that, you need to run client side program and send the request.
- As soon as the request is sent from the client end, server will respond back. Below snapshot represents the same.

Snap 01

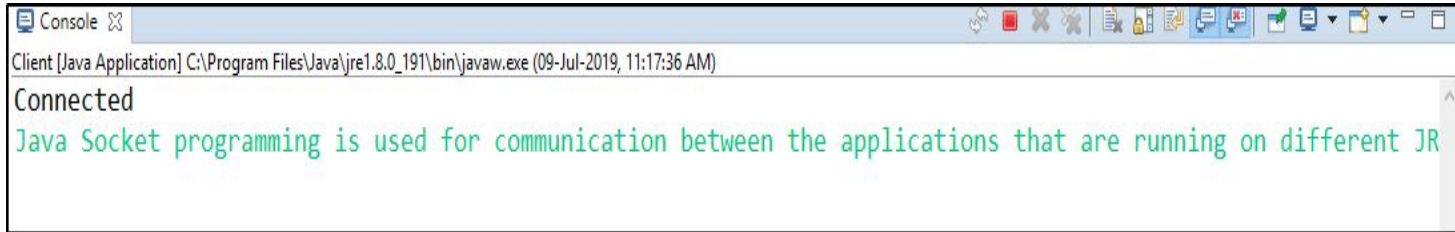
1. When you run the server side script, it will start and wait for the client To gets started.



```
Console [X]  
Server [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (08-Jul-2019, 3:34:37 PM)  
Server started  
Waiting for a client ...
```

Snap 02

2. Next, the client will get connected and inputs the request in the form of a string.

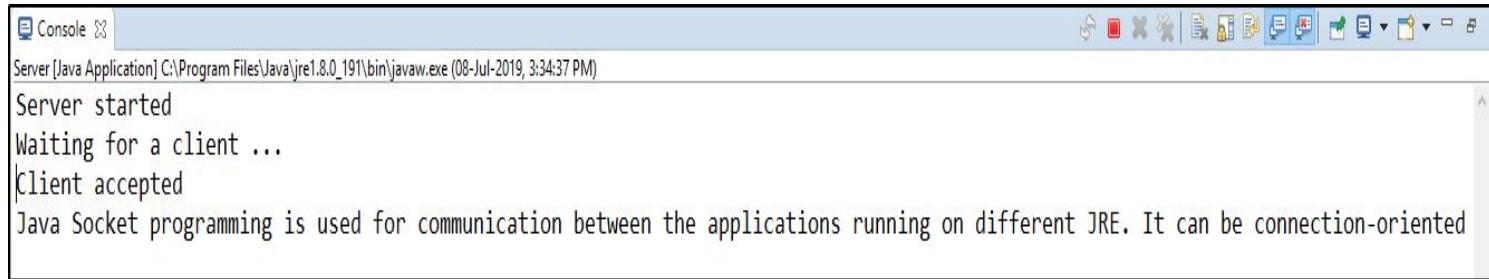


The screenshot shows a Java console window titled "Console". The title bar includes standard Windows window controls (minimize, maximize, close) and a search icon. The console text shows the client's execution path and time: "Client [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (09-Jul-2019, 11:17:36 AM)". Below this, the word "Connected" is displayed. The main output is a green-colored string: "Java Socket programming is used for communication between the applications that are running on different JR".

```
Client [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (09-Jul-2019, 11:17:36 AM)
Connected
Java Socket programming is used for communication between the applications that are running on different JR
```


Snap 03

3. When the client sends the request, the Server will respond back.

A screenshot of a Java console window. The title bar reads 'Console'. The text inside the console shows the server starting, waiting for a client, and then accepting a client connection. The text is as follows:

```
Server [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (08-Jul-2019, 3:34:37 PM)  
Server started  
Waiting for a client ...  
Client accepted  
Java Socket programming is used for communication between the applications running on different JRE. It can be connection-oriented
```

End Of Java Socket Programming.. !

Thank You..!