Networking-Java

NILESH BHANDARE

Agenda

- Basic Networking – Java
- Connection-oriented communication
  - Server Socket Programming – One Way
  - Server Socket Programming – Two Way
- Connection-less communication

Networking

- The process of connecting the resources together to share the data is called **networking**

- **java.net package** contain number of classes , using that classes we are able to connection between devices

- java.net package provide support for TCP and UDP protocols

- sender/ client & receiver/server

Cont..

- Categories of networking
  - Peer to peer networking
    - Client sometime behave as a server and server sometime behave like a client
  - Client server networking
    - Client behave as a client and server behave as a server

Basic Terminology

- IP Address
- URL
- Protocol
- Port Number
- MAC Address
- Connection Oriented and Connection Less Protocol
- Socket

Assignment 1

- Retrieve IP address of a given site.

Fetch IP

```
package NET;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.InetAddress;
public class IPDEMO1
{
public static void main(String[] args) throws Exception
    {
   BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
   System.out.println("Enter The Site Name:");
   String sitename =br.readLine();

   InetAddress in =InetAddress.getByName(sitename);
   System.out.println("IP of the "+in);
    }
}
```

Explore

- Java InetAddress Class Methods
  - **getAllByName**
  - **getLocalHost**
  - **equals()** – To check two IPs are equal or not by using
  - **getHostAddress**

https://www.tutorialandexample.com/inetaddress-class/
https://docs.oracle.com/javase/7/docs/api/java/net/InetAddress.html
https://www.geeksforgeeks.org/java-net-inetaddress-class-in-java/

URL Class

- URL is a class present in Java.net package
- by using URL class - it is possible to access information present in world wide web
- URL contain information
  - used protocol
  - server name
  - port number
  - filename or directory name
- Create Object-
- `URL url=new URL ("https://mitaoe.ac.in:8080/index”);`

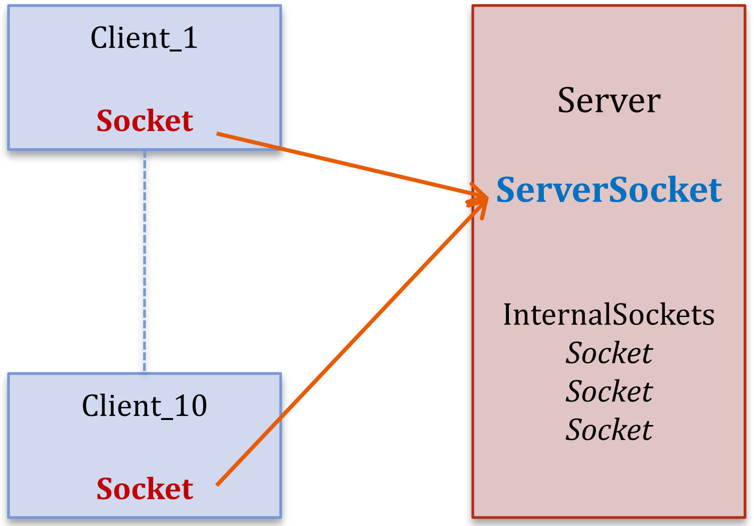
URL DEMO

```
package NET;
import java.net.URL;

public class URLDEMO
{

public static void main(String[] args) throws Exception
   {
   URL url=new URL ("https://mitaoe.ac.in:8080/index");
   System.out.println("Protocol is: "+url.getProtocol());
   System.out.println("Host Name is: "+url.getHost());
   System.out.println("Port Number is: "+url.getPort());
   System.out.println("Path is: "+url.getPath());
   System.out.println(url);

   }
}
```

Connection Oriented

- Using combination of two protocol TCP and IP
- TCP - transfer data in the form of packets between source and destination
- IP - finding IP address of a particular system
- To achieve connection oriented networking need to use
  - socket
  - Server socket
-

Socket- Server Programming in Java



1. Server need to initiate serversocket object using port number
2. Server invoke Accept() method of serversocket class to establish connection with client
3. Client create socket object using Server Name and Port Number-client site
4. Attempt by client to connect

Refer- https://www.youtube.com/watch?v=Tks7O0z1XTE

Two way chat application

https://www.geeksforgeeks.org/establishing-the-two-way-communication-between-server-and-client-in-java/

UDP- **Java.net.DatagramSocket**

- Java **DatagramSocket** and **DatagramPacket** classes are used for connection-less socket programming using the UDP instead of TCP**.**
- Java **DatagramSocket class** represents a connection-less socket for sending and receiving datagram packets
- A datagram is basically an information but there is no guarantee of its content, arrival or arrival time**.**
  - **DatagramSocket() throws SocketEeption:** it creates a datagram socket and binds it with the available Port Number on the localhost machine.
  - **DatagramSocket(int port) throws SocketEeption:** it creates a datagram socket and binds it with the given Port Number.
  - **DatagramSocket(int port, InetAddress address) throws SocketEeption:** it creates a datagram socket and binds it with the specified port number and host address.

To learn more about **Java.net.DatagramSocket:** https://www.geeksforgeeks.org/java-net-datagramsocket-class-java/

- **Java DatagramPacket** is a message that can be sent or received. It is a data container. If you send multiple packet, it may arrive in any order. Additionally, packet delivery is not guaranteed.
  - **Receive- DatagramPacket(byte[] barr, int length):** it creates a datagram packet. This constructor is used to receive the packets.
  - **Send- DatagramPacket(byte[] barr, int length, InetAddress address, int port):** it creates a datagram packet. This constructor is used to send the packets.

UDP- **Java.net.DatagramSocket**

- **Java DatagramSocket Class**
  - `void send(DatagramPacket p)` - It sends the datagram packet from the socket.
  - `void receive(DatagramPacket p)` - It receives the datagram packet from the socket.
  - `SocketAddress getSocketAddress()` - It gets the SocketAddress (IP address + port number) of the remote host that the packet is being sent to or is coming from.
  - `disconnect()` : Disconnects the socket. If the socket is not connected, then this method has no effect.
  - `connect()` : Connects to the specified address and port.
  - `getBytes()` method does the encoding of string into the sequence of bytes and keeps it in an array of bytes.

For additional methods refer- *https://www.javatpoint.com/DatagramSocket-and-DatagramPacket*

Connection less Protocol - *Demonstration*

Refer- https://www.youtube.com/watch?v=uhBAizUFBFg&t=9s

Thank You

Post Survey| Multi-threading & Networking

https://forms.gle/7dVnNUJf881GMUEo9