Article on Java Networking

**Java networking** is the concept of connecting two or more computing devices so that we can share resources .Networking provides software management .Java networking provides software management .Java networking support the concept of Socket . A socket identifies an endpoint in a network. The creation of Socket object implicitly establishes a connection between the client and server.

The following are the packages and its maintain classes that help in accessing Sockets:

1.Java.net.packages

2.ServerSocket

3.Socket

The java.net package contains all the classes required to create network enable applications. Thus, the Server Socket are for Server .and the Socket class is for Client.

**The supported protocols are :**

1.TCP

2.UDP

3. Raw IP

**Client Side Programming:**

To write a client application two things are needed:

1.IP address of Server

2.TCP Port

**Server Side Programming:**

To write a server application two things are needed:

1.A Server Socket which waits for the client request

2.A plain old Socket socket to use for communication with the client

Example on client:

package client;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Socket;

import javax.swing.JFrame;

public class Client {

public Client() {

try {

Socket s = new Socket("127.0.0.1", 1103);

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

String data = br.readLine();

System.out.println(data);

PrintWriter pw = new PrintWriter(s.getOutputStream());

pw.println("hello");

pw.flush();

} catch (Exception ex) {

System.out.println("sorry");

}

}

public static void main(String[] args) {

new Client();

}

}

Example on Server:

package server;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.ServerSocket;

import java.net.Socket;

public class Server {

public Server() {

try{

ServerSocket server=new ServerSocket(1103);

System.out.println("Server created");

Socket s=server.accept();

System.out.println("Client"+s.getInetAddress());

PrintWriter pw=new PrintWriter(s.getOutputStream());

pw.println("hi");

pw.flush();

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

String data=br.readLine();

System.out.println(data);

}catch(Exception ex){

System.out.println("Sorry");

}

}

public static void main(String[] args) {

new Server(); }}