Sockets

Java Class Library provides two main classes to interact between programs using socket:

* Socket
* ServerSocket

ServerSocket

Creates

Socket

Socket

**Connection**

**request**

**interaction**

**Client**

**Server**

**Example server**:

import java.io.\*;  
import java.net.\*;  
  
public class EchoServer {  
    private static final int PORT = 34522;  
  
    public static void main(String[] args) {  
        try (ServerSocket server = new ServerSocket(PORT)) {  
            while (true) {  
                try (  
                    Socket socket = server.accept(); // accepting a new client  
                    DataInputStream input = new DataInputStream(socket.getInputStream());  
                    DataOutputStream output = new DataOutputStream(socket.getOutputStream())  
                ) {  
                    String msg = input.readUTF(); // reading a message  
                    output.writeUTF(msg); // resend it to the client  
                }  
            }  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}

Example Client:

import java.io.\*;  
import java.net.Socket;  
import java.util.Scanner;  
  
public class EchoClient {  
    private static final String SERVER\_ADDRESS = "127.0.0.1";  
    private static final int SERVER\_PORT = 34522;  
  
    public static void main(String[] args) {  
        try (  
            Socket socket = new Socket(SERVER\_ADDRESS, SERVER\_PORT);  
            DataInputStream input = new DataInputStream(socket.getInputStream());  
            DataOutputStream output  = new DataOutputStream(socket.getOutputStream())  
        ) {  
            Scanner scanner = new Scanner(System.in);  
            String msg = scanner.nextLine();  
              
            output.writeUTF(msg); // sending message to the server  
            String receivedMsg = input.readUTF(); // response message  
  
            System.out.println("Received from server: " + receivedMsg);  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}

It’s only one message!

For handle 5 message we have to make modification on server and client

Client:

for (int i = 0; i < 5; i++) {  
    Scanner scanner = new Scanner(System.in);  
    String msg = scanner.nextLine();  
  
    output.writeUTF(msg);  
    String receivedMsg = input.readUTF();  
  
    System.out.println(receivedMsg);  
}

Server:

for (int i = 0; i < 5; i++) {  
    String msg = input.readUTF(); // reading the next client message  
    output.writeUTF(msg); // resend it to the client  
}

Multithreaded Server

import java.io.\*;  
import java.net.ServerSocket;  
import java.net.Socket;

public class EchoServer {  
    private static final int PORT = 34522;  
  
    public static void main(String[] args) {  
        try (ServerSocket server = new ServerSocket(PORT)) {  
            while (true) {  
                Session session = new Session(server.accept());  
                session.start(); // it does not block this thread  
            }  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}  
  
class Session extends Thread {  
    private final Socket socket;  
  
    public Session(Socket socketForClient) {  
        this.socket = socketForClient;  
    }  
  
    public void run() {  
        try (  
            DataInputStream input = new DataInputStream(socket.getInputStream());  
            DataOutputStream output = new DataOutputStream(socket.getOutputStream())  
        ) {  
            for (int i = 0; i < 5; i++) {  
                String msg = input.readUTF();  
                output.writeUTF(msg);  
            }  
            socket.close();  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }  
}