

**Instruction:**

1. Make a folder named anything whatever you want.
2. Make two files named ***Server.java*** and ***Client.java*** in the folder.
3. Copy the code to the files.
4. Open a terminal in the folder.
5. Compile them in the terminal by the following command

* javac Server.java Client.java

1. Run Server class by the command:

* java Server

1. Open another termial and run the Client class

* java Client

**Server code**

import java.io.\*;

import java.net.\*;

import java.time.LocalDate;

import java.time.LocalTime;

public class Server {

public static void main(String[] args) {

try (ServerSocket serverSocket = new ServerSocket(12345)) {

System.out.println("Server is running and waiting for clients...");

while (true) {

Socket clientSocket = serverSocket.accept();

System.out.println("Client connected: " + clientSocket.getInetAddress().getHostAddress());

new Thread(() -> handleClient(clientSocket)).start();

}

} catch (IOException e) {

System.err.println("Error in server: " + e.getMessage());

}

}

private static void handleClient(Socket clientSocket) {

try (

BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true)

) {

String clientMessage;

while ((clientMessage = in.readLine()) != null) {

System.out.println("Received from client: " + clientMessage);

switch (clientMessage) {

case "Hi":

out.println("Hello");

break;

case "Date":

out.println(LocalDate.now());

break;

case "Time":

out.println(LocalTime.now());

break;

case "IP":

out.println(clientSocket.getInetAddress().getHostAddress());

break;

default:

out.println("Invalid Command");

}

}

} catch (IOException e) {

System.err.println("Error handling client: " + e.getMessage());

} finally {

try {

clientSocket.close();

} catch (IOException e) {

System.err.println("Error closing client socket: " + e.getMessage());

}

}

}

}

**Client Code**

import java.io.\*;

import java.net.\*;

public class Client {

public static void main(String[] args) {

try (Socket socket = new Socket("localhost", 12345)) {

System.out.println("Connected to the server.");

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

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

String userInput;

while (true) {

System.out.print("Enter a command (Hi, Date, Time, IP, or exit): ");

userInput = console.readLine();

if ("exit".equalsIgnoreCase(userInput)) {

System.out.println("Closing connection...");

break;

}

out.println(userInput);

String response = in.readLine();

System.out.println("Server response: " + response);

}

} catch (IOException e) {

System.err.println("Error in client: " + e.getMessage());

}

}

}