**Practical No : 20**

**Aim :** Program to send and receive Data from Server

**Performed By:** Yogesh Gavande

**Class:** BCA-III SEM-V

**Date:** 05/10/2024

**Code :**

**MainActivity.java**

package com.example.Server\_aplication;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.io.PrintWriter;  
import java.net.Socket;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView tvMessages;  
 private EditText etMessage;  
 private Button btnSend;  
 private Socket socket;  
 private PrintWriter output;  
 private BufferedReader input;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 tvMessages = findViewById(R.id.tvMessages);  
 etMessage = findViewById(R.id.etMessage);  
 btnSend = findViewById(R.id.btnSend);  
  
 btnSend.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String message = etMessage.getText().toString();  
 new SendMessageTask().execute(message);  
 }  
 });  
  
 new ConnectTask().execute();  
 }  
  
 private class ConnectTask extends AsyncTask<Void, Void, Void> {  
 @Override  
 protected Void doInBackground(Void... voids) {  
 try {  
 socket = new Socket("10.0.2.2",9876);  
 output = new PrintWriter(socket.getOutputStream(), true);  
 input = new BufferedReader(new InputStreamReader(socket.getInputStream()));  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 tvMessages.setText("Connected to server");  
 }  
 });  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
 }  
  
 private class SendMessageTask extends AsyncTask<String, Void, String> {  
 @Override  
 protected String doInBackground(String... strings) {  
 String message = strings[0];  
 output.println(message);  
 try {  
 return input.readLine();  
 } catch (Exception e) {  
 e.printStackTrace();  
 return null;  
 }  
 }  
  
 @Override  
 protected void onPostExecute(String response) {  
 if (response != null) {  
 tvMessages.append("\nServer: " + response);  
 }  
 }  
 }  
}

**activity\_main.xml**

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/tvMessages"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Messages"  
 android:layout\_marginTop="16dp"/>  
  
 <EditText  
 android:id="@+id/etMessage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/tvMessages"  
 android:hint="Enter Message"/>  
  
 <Button  
 android:id="@+id/btnSend"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/etMessage"  
 android:text="SEND"/>

</RelativeLayout>

**AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.INTERNET"/>  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>  
  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Server\_aplication"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

**SimpleServer.java ( Server Side)**

import java.io.\*;  
import java.net.\*;  
  
public class SimpleServer {  
 public static void main(String[] args) {  
 int port = 9876; // Port number

try (ServerSocket serverSocket = new ServerSocket(port)) {  
 System.*out*.println("Server is listening on port " + port);  
  
 while (true) {  
 Socket socket = serverSocket.accept();  
 System.*out*.println("New client connected");  
  
 new ServerThread(socket).start();  
 }  
 } catch (IOException ex) {  
 System.*out*.println("Server exception: " + ex.getMessage());  
 ex.printStackTrace();  
 }  
 }  
}  
  
class ServerThread extends Thread {  
 private Socket socket;  
  
 public ServerThread(Socket socket) {  
 this.socket = socket;  
 }  
  
 public void run() {  
 try (InputStream input = socket.getInputStream();  
 BufferedReader reader = new BufferedReader(new InputStreamReader(input));  
 OutputStream output = socket.getOutputStream();  
 PrintWriter writer = new PrintWriter(output, true)) {  
  
 String text;  
  
 while ((text = reader.readLine()) != null) {  
 System.*out*.println("Received: " + text);  
 writer.println("Server: " + text);  
  
 if ("exit".equalsIgnoreCase(text)) {  
 System.*out*.println("Client disconnected");  
 break;  
 }  
 }  
 } catch (IOException ex) {  
 System.*out*.println("Server exception: " + ex.getMessage());  
 ex.printStackTrace();  
 }  
 }  
}

**Main.java**

public class Main {  
 public static void main(String[] args) {  
 System.*out*.printf("Hello and welcome!");  
  
 for (int i = 1; i <= 5; i++) {  
<icon src="AllIcons.Debugger.Db\_set\_breakpoint"/> breakpoint  
actionId="ToggleLineBreakpoint"/>.  
 System.*out*.println("i = " + i);  
 }  
 }  
}

**Output : Client Side**





**Output : Server Side**

