Advanced Network Programming Lab using Java

Angelos Stavrou

Table of Contents

A simple Java Client

```
import java.io.*;
import java.net.*;

public class ISAClient
{
    public static void main(String[] args) throws Exception
    {
        String message;
        String returnmessage;
        BufferedReader keyboard =
            new BufferedReader(new InputStreamReader(System.in));
        //server has to be listening to this port
        Socket mysock = new Socket("localhost",19000);
        DataOutputStream out = new DataOutputStream( mysock.getOutputStream());
       BufferedReader in = new BufferedReader(new InputStreamReader(mysock.getInputStream()));
        message = keyboard.readLine();
        out.writeBytes(message + "\n");
        returnmessage = in.readLine();
        System.out.println("Server Said: " + returnmessage);
        mysock.close();
    }
}
```

A simple Java Server

```
import java.io.*;
import java.net.*;

public class ISAServer
{
    public static void main(String args[]) throws Exception
    {
        String message;
        String messagereturn;
        ServerSocket serversock = new ServerSocket(19000); //can be any port
        while(true)
        {
            Socket connsock = serversock.accept();
            InputStreamReader instr =  new InputStreamReader(connsock.getInputStream());
            DataOutputStream outstr = new DataOutputStream(connsock.getOutputStream());
            BufferedReader in = new BufferedReader(instr);
            message = in.readLine();
            messagereturn = "You sent this to server: " +message.toUpperCase() + "\n";
            outstr.writeBytes(messagereturn);
        }
    }
}
```

An advanced Java Client

(example from http://java.sun.com/developer/onlineTraining/Programming/BasicJava2/socket.html)

```
import java.awt.Color;
import java.awt.BorderLayout;
import java.awt.event.*;
import javax.swing.*;

import java.io.*;
import java.net.*;

class SocketClient extends JFrame
        implements ActionListener {

   JLabel text, clicked;
   JButton button;
   JPanel panel;
   JTextField textField;
   Socket socket = null;
   PrintWriter out = null;
   BufferedReader in = null;

   SocketClient(){ //Begin Constructor
     text = new JLabel("Text to send over socket:");
     textField = new JTextField(20);
     button = new JButton("Click Me");
     button.addActionListener(this);

     panel = new JPanel();
     panel.setLayout(new BorderLayout());
```

```
     panel.setBackground(Color.white);
     getContentPane().add(panel);
     panel.add("North", text);
     panel.add("Center", textField);
     panel.add("South", button);
   } //End Constructor

  public void actionPerformed(ActionEvent event){
     Object source = event.getSource();

     if(source == button){
//Send data over socket
          String text = textField.getText();
          out.println(text);
     textField.setText(new String(""));
//Receive text from server
       try{
     String line = in.readLine();
          System.out.println("Text received :" + line);
       } catch (IOException e){
    System.out.println("Read failed");
           System.exit(1);
       }
     }
  }

  public void listenSocket(){
//Create socket connection
     try{
       socket = new Socket("kq6py", 4444);
       out = new PrintWriter(socket.getOutputStream(), true);
       in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
     } catch (UnknownHostException e) {
```

```
        System.out.println("Unknown host: kq6py.eng");
        System.exit(1);
      } catch  (IOException e) {
        System.out.println("No I/O");
        System.exit(1);
      }
  }

   public static void main(String[] args){
        SocketClient frame = new SocketClient();
    frame.setTitle("Client Program");
        WindowListener l = new WindowAdapter() {
                public void windowClosing(WindowEvent e) {
                        System.exit(0);
                }
        };

        frame.addWindowListener(l);
        frame.pack();
        frame.setVisible(true);
    frame.listenSocket();
  }
}
```

An advanced Java Server

(example from http://java.sun.com/developer/onlineTraining/Programming/BasicJava2/socket.html)

```
import java.awt.Color;
import java.awt.BorderLayout;
import java.awt.event.*;
import javax.swing.*;

import java.io.*;
import java.net.*;

class SocketServer extends JFrame
       implements ActionListener {

   JButton button;
   JLabel label = new JLabel("Text received over socket:");
   JPanel panel;
   JTextArea textArea = new JTextArea();
   ServerSocket server = null;
   Socket client = null;
   BufferedReader in = null;
   PrintWriter out = null;
   String line;

   SocketServer(){ //Begin Constructor
     button = new JButton("Click Me");
     button.addActionListener(this);

     panel = new JPanel();
     panel.setLayout(new BorderLayout());
     panel.setBackground(Color.white);
     getContentPane().add(panel);
```

```
    panel.add("North", label);
    panel.add("Center", textArea);
    panel.add("South", button);

  } //End Constructor

public void actionPerformed(ActionEvent event) {
    Object source = event.getSource();

    if(source == button){
        textArea.setText(line);
    }
}

public void listenSocket(){

  try{
    server = new ServerSocket(4444);
  } catch (IOException e) {
    System.out.println("Could not listen on port 4444");
    System.exit(-1);
  }

  try{
    client = server.accept();
  } catch (IOException e) {
    System.out.println("Accept failed: 4444");
    System.exit(-1);
  }

  try{
    in = new BufferedReader(new InputStreamReader(client.getInputStream()));
    out = new PrintWriter(client.getOutputStream(), true);
```

```
    } catch (IOException e) {
      System.out.println("Accept failed: 4444");
      System.exit(-1);
    }

    while(true){
      try{
        line = in.readLine();
//Send data back to client
        out.println(line);
      } catch (IOException e) {
        System.out.println("Read failed");
        System.exit(-1);
      }
    }
  }

  protected void finalize(){
//Clean up
     try{
        in.close();
        out.close();
        server.close();
    } catch (IOException e) {
        System.out.println("Could not close.");
        System.exit(-1);
    }
  }

  public static void main(String[] args){
        SocketServer frame = new SocketServer();
    frame.setTitle("Server Program");
        WindowListener l = new WindowAdapter() {
```

```
                public void windowClosing(WindowEvent e) {
                        System.exit(0);
                }
        };
        frame.addWindowListener(l);
        frame.pack();
        frame.setVisible(true);
    frame.listenSocket();
  }
}
```

A Multi-threaded Java Server

```
import java.awt.Color;
import java.awt.BorderLayout;
import java.awt.event.*;
import javax.swing.*;

import java.io.*;
import java.net.*;

class ClientWorker implements Runnable {
  private Socket client;
  private JTextArea textArea;

  ClientWorker(Socket client, JTextArea textArea) {
   this.client = client;
   this.textArea = textArea;
  }

  public void run(){
    String line;
    BufferedReader in = null;
    PrintWriter out = null;
    try{
      in = new BufferedReader(new InputStreamReader(client.getInputStream()));
      out = new PrintWriter(client.getOutputStream(), true);
    } catch (IOException e) {
      System.out.println("in or out failed");
      System.exit(-1);
    }

    while(true){
```

```
      try{
        line = in.readLine();
//Send data back to client
         out.println(line);
         textArea.append(line);
       } catch (IOException e) {
         System.out.println("Read failed");
         System.exit(-1);
       }
    }
  }
}

class SocketThrdServer extends JFrame{

   JLabel label = new JLabel("Text received over socket:");
   JPanel panel;
   JTextArea textArea = new JTextArea();
   ServerSocket server = null;

   SocketThrdServer(){ //Begin Constructor
     panel = new JPanel();
     panel.setLayout(new BorderLayout());
     panel.setBackground(Color.white);
     getContentPane().add(panel);
     panel.add("North", label);
     panel.add("Center", textArea);
   } //End Constructor

  public void listenSocket(){
    try{
      server = new ServerSocket(4444);
    } catch (IOException e) {
```

```
      System.out.println("Could not listen on port 4444");
      System.exit(-1);
    }
    while(true){
      ClientWorker w;
      try{
        w = new ClientWorker(server.accept(), textArea);
        Thread t = new Thread(w);
        t.start();
      } catch (IOException e) {
        System.out.println("Accept failed: 4444");
        System.exit(-1);
      }
    }
  }

  protected void finalize(){
//Objects created in run method are finalized when
//program terminates and thread exits
     try{
        server.close();
    } catch (IOException e) {
        System.out.println("Could not close socket");
        System.exit(-1);
    }
  }

  public static void main(String[] args){
        SocketThrdServer frame = new SocketThrdServer();
   frame.setTitle("Server Program");
        WindowListener l = new WindowAdapter() {
                public void windowClosing(WindowEvent e) {
                        System.exit(0);
```

```
                }
        };
        frame.addWindowListener(l);
        frame.pack();
        frame.setVisible(true);
        frame.listenSocket();
  }
}
```