

A Simple Chat Client and Server Example

COMP2396 Object-Oriented Programming and Java
Dr. Kenneth Wong

A Simple Chat Client

- Example

```
import java.io.*;
import java.net.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class SimpleChatClient {
    JTextArea incoming; // for showing messages
    JTextField outgoing; // for user inputs
    BufferedReader reader; // for receiving messages
    PrintWriter writer; // for sending messages
    Socket sock; // socket connection to the server

    public static void main(String[] args) {
        SimpleChatClient client = new SimpleChatClient();
        client.go();
    }
}
```

A Simple Chat Client

- Example

```
public void go() {  
    // sets up the network connection  
    try {  
        sock = new Socket("127.0.0.1", 5000);  
        InputStreamReader streamReader =  
            new InputStreamReader(sock.getInputStream());  
        reader = new BufferedReader(streamReader);  
        writer = new PrintWriter(sock.getOutputStream());  
        System.out.println("networking established");  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
  
    // builds the GUI  
    JFrame frame = new JFrame("Simple Chat Client");  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
}
```

A Simple Chat Client

- Example

```
// sets up a text area for showing incoming messages
incoming = new JTextArea(15, 50);
incoming.setLineWrap(true);
incoming.setWrapStyleWord(true);
incoming.setEditable(false);
JScrollPane qScroller = new JScrollPane(incoming);
qScroller.setVerticalScrollBarPolicy(
    ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
qScroller.setHorizontalScrollBarPolicy(
    ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
// sets up a text field for getting user inputs
outgoing = new JTextField(20);
// sets up a "Send" button
JButton sendButton = new JButton("Send");
sendButton.addActionListener(new SendButtonListener());
```

A Simple Chat Client

- Example

```
// layouts all the components on a panel
JPanel panel = new JPanel();
panel.add(qScroller);
panel.add(outgoing);
panel.add(sendButton);
// adds the panel to the frame
frame.add(panel, BorderLayout.CENTER);
// makes the frame visible
frame.setSize(640, 320);
frame.setVisible(true);

// starts a new thread to receive messages from the server
Thread readerThread = new Thread(new IncomingReader());
readerThread.start();

} // go
```

A Simple Chat Client

- Example

```
// inner class
public class SendButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        try {
            // sends the text in the text field to the server
            writer.println(outgoing.getText());
            writer.flush();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        // resets the text field
        outgoing.setText("");
        outgoing.requestFocus();
    }
} // SendButtonListener
```

A Simple Chat Client

- Example

```
// inner class
public class IncomingReader implements Runnable {
    public void run() {
        String message;
        try {
            // reads incoming messages from the server
            while ((message = reader.readLine()) != null) {
                System.out.println("read " + message);
                // appends the incoming message to the text area
                incoming.append(message + "\n");
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
} // IncomingReader
}
```

A Simple Chat Server

- Example

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

public class SimpleChatServer {
    ArrayList<PrintWriter> clientOutputStreams;
    ServerSocket serverSocket;

    public static void main(String[] args) {
        SimpleChatServer server = new SimpleChatServer();
        server.go();
    }

    public void go() {
        clientOutputStreams = new ArrayList<PrintWriter>();
    }
}
```


A Simple Chat Server

- Example

```
try {
    serverSocket = new ServerSocket(5000);
    while (true) {
        Socket clientSocket = serverSocket.accept();
        PrintWriter writer =
            new PrintWriter(clientSocket.getOutputStream());
        clientOutputStreams.add(writer);
        // creates a new thread to handle this new client
        Thread t = new Thread(new ClientHandler(clientSocket));
        t.start();
        System.out.println("got a connection");
    }
} catch (Exception ex) {
    ex.printStackTrace();
}
} // go
```

A Simple Chat Server

- Example

```
// inner class
public class ClientHandler implements Runnable {
    BufferedReader reader;
    Socket sock;

    public ClientHandler(Socket clientSocket) {
        try {
            sock = clientSocket;
            InputStreamReader isReader =
                new InputStreamReader(sock.getInputStream());
            reader = new BufferedReader(isReader);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

A Simple Chat Server

- Example

```
public void run() {  
    String message;  
    try {  
        while ((message = reader.readLine()) != null) {  
            System.out.println("read " + message);  
            broadcasts(message);  
        }  
    } catch (Exception ex) {    ex.printStackTrace();}  
}  
} // ClientHandler
```

A Simple Chat Server

- Example

```
public void broadcasts(String message) {  
    for (PrintWriter writer : clientOutputStreams) {  
        try {  
            writer.println(message);  
            writer.flush();  
        } catch (Exception ex) {  
            ex.printStackTrace();  
        }  
    }  
} // broadcasts  
}
```