CoderPanda

Simplest Java/J2EE Tutorials

posted on MARCH 23, 2013 by BIJOY

Socket Programming-Chat application in Java

Filed under SOCKET PROGRAMMING 16

So far we discussed about socket communication principles . Examples on **TCP communication** and **UDP communication** were also discussed. In this chapter we are discussing a console based(no GUI for this application) chat application in Java

Chat application in Java

It uses TCP socket communication .We have a server as well as a client.Both can be run in the same machine or different machines.If both are running in the machine , the adress to be given at the client side is local host address.If both are running in different machines , then in the client side we need to specify the ip address of machine in which server application is running.

Let us inspect the server code first.

The **ChatSocketServer.java** is the server application.It simply creates a serverSocket on port 3339.Once a new connection comes , it accepts that connection and Socket object will be created for that connection.Now two threads will be created .One thread is for reading from the socket and the other is writing to socket.If the connection is terminated from client side , the server also exits.

ChatSocketServer.java

```
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketException;
```

```
public class ChatSocketServer {
private ServerSocket severSocket = null;
private Socket socket = null;
private InputStream inStream = null;
private OutputStream outStream = null;

public ChatSocketServer() {
```

Translate Page To:

Search this blog

Adds

Advertisements

Categories

- DataBase (10)
- Java (81)
  - advanced Java topics (1)
  - Annotattions in java (2)
  - Core Java (60)
  - Socket Programming (11)
    - » Java Socket Programming Tutorial
    - » Java Socket Programming Using TCP
    - » Java Socket Programming Using UDP
    - » Java Socket Programming-File transfer through socket in Java
    - » Java Socket programming-Transferring directory through socket in Java
    - » Java Socket Programming-Transferring File through Socket in Java
    - » Java Socket Programming-Transferring file using UDP
    - » Java Socket programming-Transferring Java object through socket using UDP
    - » Java Socket Programming-Transferring large sized files through socket
    - » Java Socket Programming-Transferring of Java Objects through sockets
    - » **Socket Programming-Chat application in Java**
  - XML Processing in Java (7)

Translate »

```
public ChatSocketServer() {

}

public void createSocket() {
try {
ServerSocket serverSocket = new ServerSocket(3339);
while (true) {
socket = serverSocket.accept();
inStream = socket.getInputStream();
outStream = socket.getOutputStream();
System.out.println("Connected");
createReadThread();
createWriteThread();

}
} catch (IOException io) {
io.printStackTrace();
}
}

public void createReadThread() {
Thread readThread = new Thread() {
public void run() {
while (socket.isConnected()) {
try {
byte[] readBuffer = new byte[200];
int num = inStream.read(readBuffer);
if (num > 0) {
byte[] arrayBytes = new byte[num];
System.arraycopy(readBuffer, 0, arrayBytes, 0, num);
String recvedMessage = new String(arrayBytes, "UTF-8");
System.out.println("Received message :" + recvedMessage);
} else {
notify();
}
;
//System.arraycopy();

} catch (SocketException se) {
System.exit(0);

} catch (IOException i) {
i.printStackTrace();
}

}
}
};
readThread.setPriority(Thread.MAX_PRIORITY);
readThread.start();
}

public void createWriteThread() {
Thread writeThread = new Thread() {
public void run() {

while (socket.isConnected()) {
try {
```

» Default method in interface in java
Java EE (56)
Java Script (1)
node.js (1)
Spring Framework (3)
Tech News (2)
Uncategorized (1)

```
BufferedReader inputReader = new BufferedReader(new
InputStreamReader(System.in));
sleep(100);
String typedMessage = inputReader.readLine();
if (typedMessage != null && typedMessage.length() > 0) {
synchronized (socket) {
outStream.write(typedMessage.getBytes("UTF-8"));
sleep(100);
}
}/* else {
notify();
}*/
;
//System.arraycopy();

} catch (IOException i) {
i.printStackTrace();
} catch (InterruptedException ie) {
ie.printStackTrace();
}

}
}
};
writeThread.setPriority(Thread.MAX_PRIORITY);
writeThread.start();

}

public static void main(String[] args) {
ChatSocketServer chatServer = new ChatSocketServer();
chatServer.createSocket();
```

```
}
}
```

Now let us start looking into the client side code.

The ChatSocketClient.java simply creates socket connection with the specified address on port 3339.Once a connection is established, two threads are creating.One for reading from the socket and other for writing to socket.Once the server disconnects the connection, the client exists itself.

ChatSocketClient.java

```
import java.io.*;
import java.net.Socket;
import java.net.SocketException;
import java.net.UnknownHostException;
```

```
public class ChatSocketClient {
private Socket socket = null;
private InputStream inStream = null;
private OutputStream outStream = null;

public ChatSocketClient() {

}
```

```
}

public void createSocket() {
try {
socket = new Socket("localHost", 3339);
System.out.println("Connected");
inStream = socket.getInputStream();
outStream = socket.getOutputStream();
createReadThread();
createWriteThread();
} catch (UnknownHostException u) {
u.printStackTrace();
} catch (IOException io) {
io.printStackTrace();
}
}

public void createReadThread() {
Thread readThread = new Thread() {
public void run() {
while (socket.isConnected()) {

try {
byte[] readBuffer = new byte[200];
int num = inStream.read(readBuffer);

if (num > 0) {
byte[] arrayBytes = new byte[num];
System.arraycopy(readBuffer, 0, arrayBytes, 0, num);
String recvedMessage = new String(arrayBytes, "UTF-8");
System.out.println("Received message :" + recvedMessage);
}/* else {
// notify();
}*/
;
//System.arraycopy();
}catch (SocketException se){
System.exit(0);

} catch (IOException i) {
i.printStackTrace();
}

}
}
};
readThread.setPriority(Thread.MAX_PRIORITY);
readThread.start();
}

public void createWriteThread() {
Thread writeThread = new Thread() {
public void run() {
while (socket.isConnected()) {

try {
BufferedReader inputReader = new BufferedReader(new
InputStreamReader(System.in));
sleep(100);
```

```
sleep(100);
String typedMessage = inputReader.readLine();
if (typedMessage != null && typedMessage.length() > 0) {
synchronized (socket) {
outStream.write(typedMessage.getBytes("UTF-8"));
sleep(100);
}
}
;
//System.arraycopy();

} catch (IOException i) {
i.printStackTrace();
} catch (InterruptedException ie) {
ie.printStackTrace();
}

}
}
};
writeThread.setPriority(Thread.MAX_PRIORITY);
writeThread.start();
}
```

```
public static void main(String[] args) throws Exception {
ChatSocketClient myChatClient = new ChatSocketClient();
myChatClient.createSocket();
/*myChatClient.createReadThread();
myChatClient.createWriteThread();*/
}
}
```

Output

Run the ChatSocketServer.java and then the ChatSocketClient.java.Once both hot connected, connected message will be displayed on the console.Now type messages on each console and press enter button.Messages will be transmitted throgh socket.

Output of ChatSocketServer.java

Connected

Received message :haai

I am server

who are u ?

Received message :i am client ...How r u ?

Output of ChatSocketClient,java

Connected

haai

Received message :I am server

Received message :who are u ?

i am client ...How r u ?

See related topics:

Networking with Java – overview

TCP communication with Java

UDP communication with Java

Tagged with Chat application in Java Chat application using TCP Socket

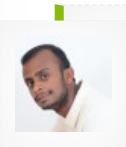
← Previous Next →

16 thoughts on "Socket Programming-Chat application in Java"

nahid *July 15, 2013 at 2:44 pm*

excuse me i want to know how can i connect these client and severs in a form i java to make it graphical?

Reply



Bijoy *July 15, 2013 at 6:38 pm*

I think you are trying to make GUI for this app. For that you can enhance this application using GUI techniques like Swing

Reply

Zeet *February 5, 2015 at 4:42 pm*

Hi Nahid,
Did you get the solution for the same , where u can use the things with form . If yes please share your experience. We are wondering for the same.

Reply

Shubham *May 11, 2014 at 10:12 pm*

thanx dude...!! it really worked without any error..!!!

Reply

shivakumar *May 16, 2014 at 4:45 pm*

Thanks a lot for giving valuable and working code.. 🙂

Reply

thomprl *June 20, 2014 at 10:36 pm*

I'm trying to use your code to connect to a VB.NET socket program. Everything works good but for some reason if I send a text shorter than the previous text it shows text from the previous one sent.

Java Send TEST12345
VB Recv TEST12345

Translate »