CyberAsylum

For all those who are mad about computing…

Object Serialization over Networks in Java

Posted on **January 25, 2011**

When building a client – server based application, we need to send objects between the client and the server applications. And java provides two great mechanisms to achieve this: Object Serialization and Sockets. Once you know the basics of these two mechanisms, putting them together is not very hard. However there are some pitfalls that you need to avoid.

For this example I will make a small application.

> **The client application**: will connect to the server and pass objects to the server. The passed objects will contain a String which the user entered on the console.
>
> **The server application**: will echo the messages back to the client.

I assume you know a bit about object serialization and socket programming. But to give a brief introduction about serializing objects:

> The object needs to implement the ‘Serializable’ interface. If you are going to serialize an object which is composed of many other objects, all those nested objects need to implement the ‘Serializable’ interface as well. If you happen to get a “java.io.NotSerializableException”, then it means you are trying to serialize an object that does not implement the Serializable interface. Most of Java’s objects do implement this interface.
>
> When you have a ‘Serializable’ object with you, you can write it using the ObjectOutputStream.

```
ObjectOutputStream out = new ObjectOutputStream(/*Some output Stram*/);
out.writeObject(/*Some Serializable Object*/);
out.close();
```

To read the object the ObjectInputStream can be used:

```
ObjectInputStream in = new ObjectInputStream(/*Some input Stram*/);
/*Some Serializable object*/  =  in.readObject();
in.close();
```

Now let’s get into the example. I have created a ‘Message’ object which will be used to communicate between the server and the client. Basically the server and client will send ‘Message’ objects to each other.

Message class:

```
import java.io.Serializable;

/**
 *
 * @author Janith (http://cyberasylum.wordpress.com/)
 */
public class Message implements Serializable{

    private String message;

    public Message(String message) {
        this.message = message;
    }

    public String getMessage(){
        return message;
    }
}
```

The server application will wait for a connection, read the objects from the socket and reply,

Server Class:

```
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;

/**
 *
 * @author Janith (http://cyberasylum.wordpress.com/)
 */
public class Server {

    private static final int PORT = 5000;

    public static void main(String[] args) {
        ServerSocket serverSocket = null;
        try {
            //Creates a new server socket with the given port number
            serverSocket = new ServerSocket(PORT);
        } catch (IOException ex) {
```

```java
            System.out.println("Error occured while creating the server socket");
            return;
        }

        Socket socket = null;
        try {
            //Waits untill a connection is made, and returns that socket
            socket = serverSocket.accept();
        } catch (IOException ex) {
            System.out.println("Error occured while accepting the socket");
            return;
        }
        //Now we have established the a connection with the client
        System.out.println("Connection created, client IP: " + socket.getInetAddress());
        ObjectInputStream in = null;
        ObjectOutputStream out = null;
        while (true) {
            try {
                if (in == null) {
                    in = new ObjectInputStream(socket.getInputStream());
                }
                Message message = (Message) in.readObject();
                System.out.println("Client said: " + message.getMessage());

                //Send a reply to the client
                if (out == null) {
                    out = new ObjectOutputStream(socket.getOutputStream());
                }
                out.writeObject(new Message("Message recieved: " + message.getMessage()));
                out.flush();
            } catch (Exception ex) {
                System.out.println("Error: " + ex);
            }
        }
    }
}
```

Client Class:

```java
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.net.Socket;
import java.util.Scanner;

/**
 *
 * @author Janith (http://cyberasylum.wordpress.com/)
 */
public class Client {

    private static final String SERVER_IP = "127.0.0.1";
    private static final int SERVER_PORT = 5000;

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);   //to read text from the console
        Socket socket = null;
        try {
            socket = new Socket(SERVER_IP, SERVER_PORT);
            System.out.println("Connected to server!");
        } catch (Exception ex) {
            System.out.println("Error connecting to server: " + ex.getMessage());
        }

        ObjectInputStream in = null;
        ObjectOutputStream out = null;
        while (true) {
            try {
                if (out == null) {
                    out = new ObjectOutputStream(socket.getOutputStream());
                }

                //read a string
                System.out.println("Enter a message: ");
                String str = scanner.next();

                //send it to server
                out.writeObject(new Message(str));
                out.flush();

                //get the reply from the server
                if (in == null) {
                    in = new ObjectInputStream(socket.getInputStream());
                }
                Message message = (Message) in.readObject();
                System.out.println("Server said: " + message.getMessage());

            } catch (Exception ex) {
                System.out.println("Error: " + ex);
            }
        }
    }
}
```

You can download the Netbeans project folder of this project from here (http://dl.dropbox.com/u/7290180/Serialization.rar).

Run the server part first and then run the client program.

A word of warning (and perhaps the most important part of this post)

The constructor of the ObjectInputStream 'wait's until the ObjectOutputStream of the other side to write a header. So if you try to construct the ObjectInputStream at the very beginning of both client and server program, you will end up in an ugly dead lock! Basically the client side will wait until the server to create its end of the stream, and the server side will wait until the client to create its end of the stream, and both sides will wait forever!

So, how do we solve this?

You should decide which side initiates the connection. i.e. you should decide which side would send an object first, and the other side should be prepared to read that. In the above example, the client sends the first message (so on the client side, first the ObjectOutputStream is created and on the server side, the ObjectInputStream is created). So I guess it would be a good rule of thumb to create the object input and output streams just before you use them.

Finally, in the above example the server is able to handle only one connection at a time, and that would be very inefficient for a server. So I have uploaded a better version of the above program where the server can handle multiple connections simultaneously using threads. Click here (http://dl.dropbox.com/u/7290180/SimpleClientServer.rar) to download the Netbeans project folder of the improved client server project.

I know this is a very long post, and hope I didn't make it boring 🙂 . Hope you will find this useful.

Share this: Facebook 11 0 Email Print Twitter 1

Like this: ★Like 2 bloggers like this.

This entry was posted in **Uncategorized** and tagged **Java**, **Programming** by **Janith**. Bookmark the **permalink [http://cyberasylum.wordpress.com/2011/01/25/object-serialization-over-networks-in-java/]** .

11 THOUGHTS ON "OBJECT SERIALIZATION OVER NETWORKS IN JAVA"

Pasindu
on **January 25, 2011 at 6:18 pm** said:

great work machan!! will be useful

Gayashan
on **January 25, 2011 at 7:44 pm** said:

Thanx a lot for the tutorial mchn….

Sunimal
on **January 25, 2011 at 9:26 pm** said:

Really useful stuff machan! keep it up!

Janith
on **January 25, 2011 at 9:35 pm** said:

Thanks guys!!! Glad to be of help

Amila
on **January 26, 2011 at 9:06 pm** said:

Really useful buddy. Thanks a lot

Vathsala
on **January 28, 2011 at 7:47 pm** said:

Thumbs up!!!!

javarevisited
on **January 28, 2011 at 8:48 pm** said:

Hi,

Just to add while writing Serializable class , its good practice to put a Seriazlizable alert in file so that when some one else add a new field in the class he must ensure that field is either transient or Serializable , so that Serializability of class should not break.

Thanks
Javin
Why String is immutable in Java

lucidlts
on **March 14, 2012 at 7:50 am** said:

great work

prakash meghwal
on **March 19, 2012 at 11:12 pm** said:

i am making a chat software just like GTALK .i am having problems in transferring the objects on to the server . i have already made the coding for chatting with everyone .but i don't know how to talk to a particular person i.e chatting between two specific persons(not with everyone who is connected to the

server). while coding for common chat with everyone i have used BufferedReader and PrintWriter. now for secret chat i need to transfer the objects on to the network. i am trying yet ,but i need your support .
so please help me in overcoming this problem.

> **Janith**
> on **March 20, 2012 at 8:38 am** said:
>
> Hi Prakash,
> Interesting problem. I think what you should do is, when someone sends a private message from the client, make the client app send the message and the recipients id or name to the server.
> The server should manage queues for all the users. When you get a message for a particular user, add it to his queue. I suggest you use something like a BlockingQueue since there would be multiple threads adding messages.
> Im not exactly sure where you are stuck. If you need more help just drop me an email (slayerjay@gmail.com) and I can help you 😃

dhana
on **July 23, 2012 at 6:11 pm** said:

Nice work….keep it up….