

Traditional blocked TCP based sockets in Java

Socket server source code

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

public class Server {
    ServerSocket providerSocket;
    Socket connection = null;
    ObjectOutputStream out;
    ObjectInputStream in;
    String message;

    Server() {
    }

    void run() {
        try {
            // 1. create a socket server listening to port 8080
            providerSocket = new ServerSocket(8080, 10);
            // 2. waiting for the connection (here we are waiting until
next connection)
            connection = providerSocket.accept();
            // 3. create Input and Output streams
            out = new ObjectOutputStream(connection.getOutputStream());
            in = new ObjectInputStream(connection.getInputStream());
            // 4. socket communication
            do {
                try {
                    message = (String) in.readObject();
                    System.out.println("client>" + message);
                    if (message.equals("bye")) {
                        sendMessage("bye");
                    }
                } catch (ClassNotFoundException classnot) {
                    System.err.println("Data received in unknown
format");
                }
            } while (!message.equals("bye"));
        } catch (IOException ioException) {
            ioException.printStackTrace();
        } finally {
            // 4: close connection
            try {
                in.close();
                out.close();
            }
        }
    }
}
```

```
        providerSocket.close();
    } catch (IOException ioException) {
        ioException.printStackTrace();
    }
}

void sendMessage(String msg) {
    try {
        out.writeObject(msg);
        out.flush();
        System.out.println("server>" + msg);
    } catch (IOException ioException) {
        ioException.printStackTrace();
    }
}

public static void main(String args[]) {
    Server server = new Server();
    while (true) {
        server.run();
    }
}
}
```

Socket client source

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

public class Client {
    Socket requestSocket;
    ObjectOutputStream out;
    ObjectInputStream in;
    String message;

    Client() {
    }

    void run() {
        try {
            // 1. try to connect to the socket: localhost:8080
            requestSocket = new Socket("localhost", 8080);
            // 2. Input and Output streams
            out = new
ObjectOutputStream(requestSocket.getOutputStream());
```

```
        in = new ObjectInputStream(requestSocket.getInputStream());
        // 3: communications
        do {
            try {
                sendMessage("Hello server");
                sendMessage("bye");
                message = (String) in.readObject();
            } catch (Exception e) {
                System.err.println("data received in unknown
format");
            }
        } while (!message.equals("bye"));
    } catch (UnknownHostException unknownHost) {
        System.err.println("You are trying to connect to an unknown
host!");
    } catch (IOException ioException) {
        ioException.printStackTrace();
    } finally {
        // 4: close connection
        try {
            in.close();
            out.close();
            requestSocket.close();
        } catch (IOException ioException) {
            ioException.printStackTrace();
        }
    }
}

void sendMessage(String msg) {
    try {
        out.writeObject(msg);
        out.flush();
        System.out.println("client>" + msg);
    } catch (IOException ioException) {
        ioException.printStackTrace();
    }
}

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

From:
<https://edu.iit.uni-miskolc.hu/> - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:java_example_for_blocking_and_non-blocking_socket?rev=1678032790

Last update: **2023/03/05 16:13**

