

```

package com.sxt.tcp;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

/**
 * 熟悉流程
 * 创建服务器
 * 1. 指定端口 使用ServerSocket创建服务器
 * 2. 阻塞式等待连接accept
 * 3. 操作：输入输出流操作
 * 4. 释放资源
 * @author 江
 *
 */

public class LoginMultiServer {
    public static void main(String[] args) throws IOException {
        System.err.println("---server---");
        //1. 指定端口 使用ServerSocket创建服务器
        ServerSocket server=new ServerSocket(8888);
        //2. 阻塞式等待连接 accept
        boolean isRunning=true;
        while(isRunning) {
            Socket client =server.accept();
            System.err.println("一个客户端建立了连接");

            new Thread(new Channel(client)).start();
        }

        server.close();
    }
}

```

```
}
```

```
static class Channel implements Runnable{
```

```
    private Socket client;
```

```
    //输入流
```

```
    private DataInputStream dis;
```

```
    //输出流
```

```
    private DataOutputStream dos;
```

```
    public Channel(Socket client) {
```

```
        this.client=client;
```

```
        try {
```

```
            dis=new
```

```
DataInputStream(client.getInputStream());
```

```
            dos=new
```

```
DataOutputStream(client.getOutputStream());
```

```
        } catch (IOException e) {
```

```
            e.printStackTrace();
```

```
            release();
```

```
        }
```

```
    }
```

```
    @Override
```

```
    public void run() {
```

```
        String uname="";
```

```
        String upwd="";
```

```
        //分析
```

```
        String[] dataArray=receive().split("&");
```

```
        for(String info:dataArray) {
```

```
            String[] userInfo=info.split("=");
```

```
            if(userInfo[0].equals("uname")) {
```

```
                System.err.println("你的用户名为"+userInfo[1]);
```

```
            uname=userInfo[1];
```

```
            }else if(userInfo[0].equals("upwd")) {
```

```
                System.err.println("你的密码为"+userInfo[1]);
```

```
            upwd=userInfo[1];
```

```

    }
}
if(uname.equals("shsxt")&& upwd.equals("laopei")) { //成功
    send("登录成功, 欢迎回来");

} else { //失败
    send("用户名或密码错误");
}
release();
}

```

//接受数据

```

private String receive() {
    String datas="";
    try {
        datas = dis.readUTF();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return datas;
}

```

//发送数据

```

private void send(String msg) {
    try {
        dos.writeUTF(msg);
        dos.flush();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

//释放资源

```

private void release() {
    try {

```

```
        if(null!=dis) {
            dis.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

try {
    if(null!=dos) {
        dos.close();
    }
} catch (IOException e) {
    e.printStackTrace();
}

try {
    if(null!=client) {
        client.close();
    }
} catch (IOException e) {
    e.printStackTrace();
}

}

}
```