

程序文档

项目组：Team3

成员： 1352837 倪雨婷

1352866 罗晓丹

1352873 王刚

1352895 叶坤宇

1352918 刘旭东

一. 引言

程序编写目标：

建立一个信息通信的应用，主要包括Client以及Server两部分的功能，具体来说将实现以下基础功能：

实现用户的登录和注册；

保持用户端和服务端的不断的connection；

记录并显示客户端登录状态以及发送的信息；(Server端会忽略未登录状态下Client发送的消息)

服务端将任意客户端的信息转发到其余登录状态下的客户端；

拒绝客户端在一秒钟内发送超过5则信息到服务端；

当某一客户端发送100则总数的信息后自动退出登录，下次登录发送信息数将从0重新计数；

Server端能够记录/忽略的消息数以及自身转发的消息数，Client端能够记录该端发送的消息数以及接收的消息数；

在此基础上，实现：

分拆server，至少分拆为以下四个 server：

鉴权，消息接受，消息存储，消息转发。

二. 程序设计

本项目为基于Windows平台，采用Java语言编写，使用中间件ActiveMQ进行通信的搭建。主要设计有界面有“登录注册界面”和“客户端界面”。总体功能上分为客户端部分和服务端部分。主要功能是客户端向服务端发送信息，服务端接收信息并转发到其余处于成功登录状态的客户端上显示。

(一) Client部分：

提供登录注册功能：

首先在Server中的类User中有关于用户各自信息的定义：

这里有关的是：

```
public String UserName;  
public String Password;
```

通过遍历用户姓名，进行与输入框中的用户名的匹配，根据匹配结果再进行密码校对，从而确定登录的具体情况并在之后显示出来。

用到的组件有：

```
JLabel regUsername;  
JLabel regPassword;  
JTextField regUsernameInput;  
JPasswordField regPasswordInput;  
JButton regBtn;
```

此处“登录界面”的布局由下列代码布置：

```
username = new JLabel("用户名 :");  
password = new JLabel("密 码 :");  
usernameInput = new JTextField(10);  
passwordInput = new JPasswordField(10);  
login = new JButton("登陆");  
register = new JButton("注册");
```

具体说来，当用户点击Button“登录”的时候，会触发
`loginPanel.add(login);` 实现登录功能。

而点击“注册”按钮时候，会跳转到“注册界面”，此页面的布置与前面的登录界面类似。

登录执行之后会根据反馈结果进行相关的状态反馈：

“客户端界面”功能：

实现界面如下：



直观上可以看出，这里主要记录了登录成功次数，登录状态，登录失败次数，已发送消息数，消息的发送与接收及在线人员。向服务端执行发送的功能由Button“发送”控制。

部分功能演示结果如下：



位于“客户端”界面上面部分的四个信息，是由Client处代码对Server端所记录的数据进行调用，并进行分析解释后显示出来的结果，下方为举例说明：

Server端： (前面代码省略)

```
validLoginTime++;  
theUser.isLogin = true;  
return 200;
```

Client端： `public void ListenMsg() throws JMSEException`
`{.....}`

..... (前面代码省略)

```
if(txtMsg.getText().equals("200")){  
    feedbackDisplay.setText("登陆成功");}
```

【注：Server端返回结果200代表的是“登录成功”】

而消息发送功能【转发功能(在之后的Server端分析中将说明)】由以下代码实现:

消息发送功能: (前面代码省略)

```
sentButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        String str = msgSent.getText();
        msgNumberCount++;
        msgNumberDisplay.setText(String.valueOf(msgNumberCount));
        //msgDisplay.setText(str); //显示到消息显示框
        status=false;
        sendMsg(str, "Ericsson", false);
    }
});
.....
// 定义sendMsg() 功能函数:
public void sendMsg(String msgText, String
toipcname, boolean isLogin) {
    .....
}
```

这样便实现了将“消息发送框”中的信息文本读取并发送给到服务端的功能。

(二) Server部分:

2.1 功能实现

限制用户不得在一秒时间内发送5则以上消息:

```
if (current-user.loginDate < 1000)
{

if(user.loginRequasetTime<5)
{
    System.out.println(user.loginRequasetTime);
    user.loginRequasetTime++;
    return true;
}
}}
```

设定在用户发送消息总数小于100以及超过100时的应对:

```
sender.sendMessageNum++;
System.out.println("Server 0");
if (sender.sendMessageNum < 100) {
    System.out.println("Server 1");
    return true;
} else return false;
//超过100后退出重登录, 并将发送消息数归0
..... (前面代码省略)
sender.isLogin = false;
sender.sendMessageNum = 0;
```

监听客户端发送的消息并转发：

监听客户端：

```
public void ListenMsg() {  
    ActiveMQConnectionFactory factory = new  
    ActiveMQConnectionFactory("tcp://localhost:61616");  
    Connection connection;  
    .....}
```

这里将对客户端传递到服务端的消息进行不断的监听。

对客户端的消息进行转发：

```
public void sendMsg(String msgText,String toipcname){  
    .....  
}
```

这部分代码是对服务端将自己收到的消息按照“根据客户端状态判断”，将所收到消息转发到对应的其余客户端的功能实现。

2.2 server拆分实现

将server端拆分为四个部分，分别为鉴权，消息接收，消息存储，消息转发。

不同的功能由不同的server实现，server之间使用activeMQ通信。