


UDESC - Eng. Software
DSD

Chat cliente-servidor Java + Sockets


Lucas Dolsan

Projeto

 Fulano - x

Server: Fulano conectou

Enviar

 - x

Conectar

C ChatDisplay

```
messagesArea: JTextArea  
messageField: JTextField  
sendButton: JButton  
container: JPanel  
usernameField: JTextField  
hostField: JTextField  
portField: JTextField  
startButton: JButton  
serverConnection: ServerConnection  
  
void connect()
```

C ServerConnection

```
private Socket serverSocket;  
private SocketReader reader;  
private SocketWriter writer;  
private ChatDisplay chatDisplay;
```

C StartClient

```
void main()
```

C ServerDisplay

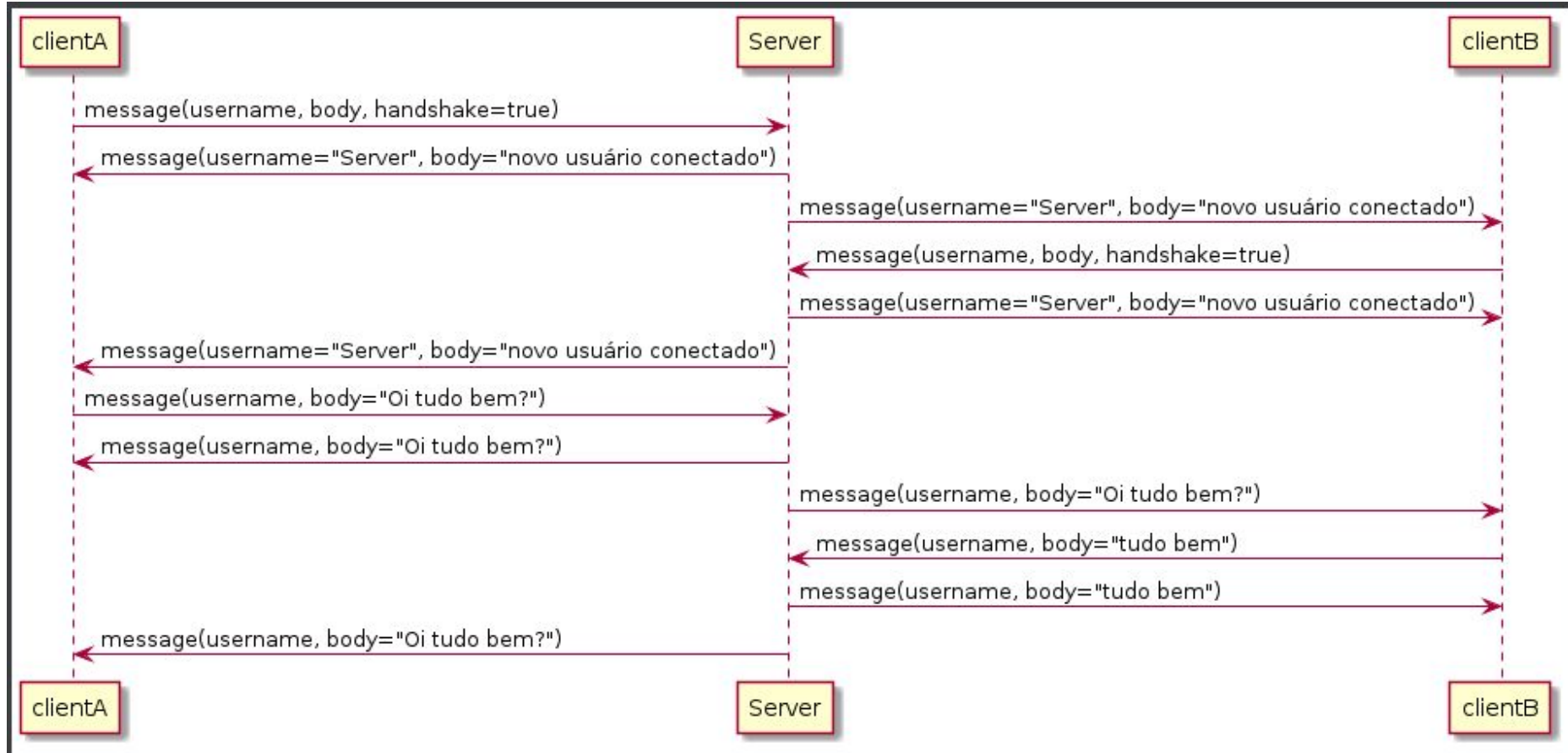
```
container: JPanel  
portField: JTextField  
maxClientField: JTextField  
startButton: JButton  
clients: List<ClientConnection>  
  
void start()
```

C ClientConnection

```
serverDisplay: ServerDisplay  
socket: Socket  
reader: SocketReader  
writer: SocketWriter  
clients: List<ClientConnection>
```

C StartServer

```
void main()
```



ServerDisplay.start()

```
1 usage
private void start() throws IOException {
    ExecutorService executor = Executors.newFixedThreadPool(this.getMaxClientCount());

    ServerSocket listener = new ServerSocket(this.getPort());

    System.out.println("Server listening at: " + this.getPort());

    while (true) {
        Socket clientSocket = listener.accept();

        ClientConnection connection = new ClientConnection(clientSocket, serverDisplay: this, clients);

        clients.add(connection);

        executor.execute(connection);
    }
}
```

ChatDisplay.connect()

1 usage 👤 Lucas Dolsan *

```
public void connect() {  
    Socket serverSocket = null;  
    try {  
        serverSocket = new Socket(this.hostField.getText(), Integer.parseInt(this.portField.getText()));  
    } catch (IOException e) {  
        throw new RuntimeException(e);  
    }  
  
    this.setTitle(getUsername());  
  
    new ServerConnection(serverSocket, chatDisplay: this).start();  
}
```

ServerConnection.run()

Thread

Lucas Dolsan *

@Override

```
public void run() {  
    String rawMessage;  
    while (true) {  
        rawMessage = this.reader.read();  
        System.out.println("raw: " + rawMessage);  
        if(!SocketReader.isMessageValid(rawMessage)) {  
            break;  
        }  
  
        Message message = Message.fromJson(new JSONObject(rawMessage));  
        this.log(message);  
        this.chatDisplay.print(message);  
    }  
    this.reader.close();  
}
```

ClientConnection.run() Thread

```
Lucas Dolsan *  
@Override  
public void run() {  
    String rawMessage;  
    while (true) {  
        rawMessage = this.reader.read();  
  
        if (!SocketReader.isValidMessage(rawMessage)) {  
            break;  
        }  
  
        Message message = Message.fromJson(new JSONObject(rawMessage));  
        this.log(message);  
  
        if (message.getHandshake()) {  
            this.username = message.getUsername();  
  
            Message handshakeResponse = this.buildHandshakeResponse(message);  
            this.writer.broadcastMessage(handshakeResponse, this.clients);  
        } else {  
            this.writer.broadcastMessage(message, this.clients);  
        }  
    }  
    this.reader.close();  
}
```


SocketWriter.broadcastMessage()

2 usages  Lucas Dolsan

```
public void broadcastMessage(Message message, ArrayList<ClientConnection> clients) {  
    for (ClientConnection clientConnection : clients) {  
  
        if(!this.shouldBroadcastToSameConnection) {  
            if (this.isSameConnection(clientConnection)) {  
                break;  
            }  
        }  
  
        this.log(message, clientConnection);  
        clientConnection.getWriter().write(message.toJson().toString());  
    }  
}
```

UDESC - Eng. Software
DSD

Chat cliente-servidor Java + Sockets

Lucas Dolsan