# 第二次功能添加

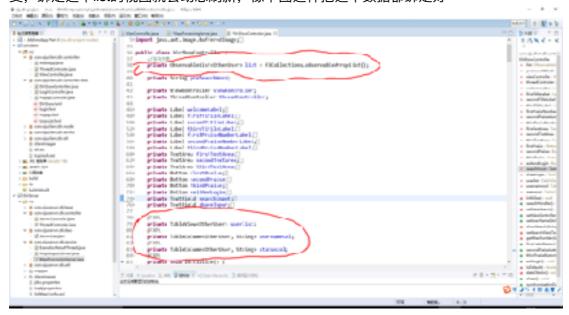
### 不用登陆就可以访问功能

在ViewController中initRootLayout方法中添加了一个判断,如果当前没用户登陆的话就自动用default登录,

### 在线人员列表功能

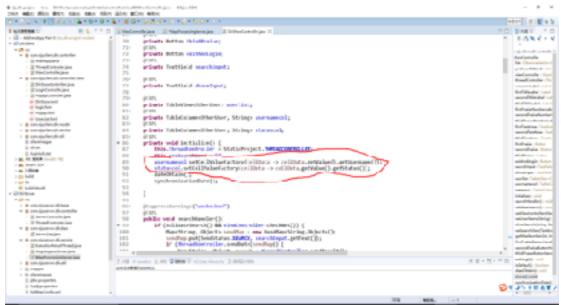
首先向后台请求在线人员列表,后台会返回一个map,这种类型的 Map<String, Map<String, String >>, map中的map是返回的人员列表,这个map也就是列表map的键是用户名,值是在线状态,这样我们就拿到了人员数据,

拿到数据以后就是渲染到页面上,我们用了TableView,里面有两个TableColumn,(布局在xml文件中,然后绑定id在对应的Controller中),首先声明一个private ObservableList<OtherUser> list = FXCollections.observableArrayList();在DirShowController中,这个东西是javafx中数据绑定用到的一个类,就是这个list数据改变,绑定这个list的视图就会动态刷新,像下图这样把这个数据都绑定好



数据绑定,就是你传给它一个东西,他获得那个东西中的值,那个东西的类型是

#### OtherUser类型的



给父层设置数据源,不然上面的那两个数据绑定不知道数据源是什么

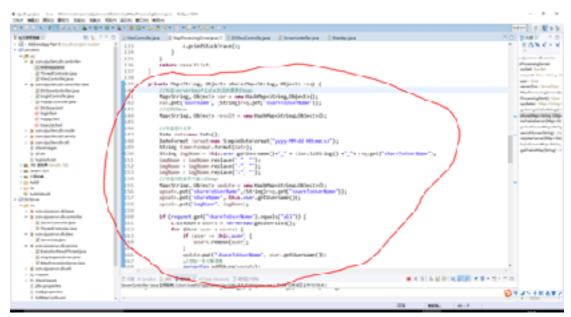


这样数据绑定就完成了,然后我们拿到数据以后,就直接添加到那个list里面就好了, javafx会自动帮我们把视图更新好,OtherUser是自己定义的bean类,

## 分享当前截图功能

点击分享按钮,触发DirShowController中的share()方法,share()方法把你要分享给谁传给ViewController的shareCare()方法,shareCare()方法通过ViewController中有个cutScreen()方法会返回BufferedImage类型的当前屏幕截图,拿到BufferedImage类型的截图,转成byte[]再转成string(BufferedImage不能传输,所以转成byte[],byte[]又不能当对象传输,所以又转了一下变成string),然后构造一个map返回,用来向后端发送,

#### 后端



拿到前端给的数据,维护数据表,然后把图片写入本地,返回成功信息

这样分享信息写到数据库中,图片写在本地的文件夹里,等到用户获得分享的时候就查表, 发现有人给他分享图片了,然后就下载对应的图片就行了,然后把表中的信息一删就ok 了,

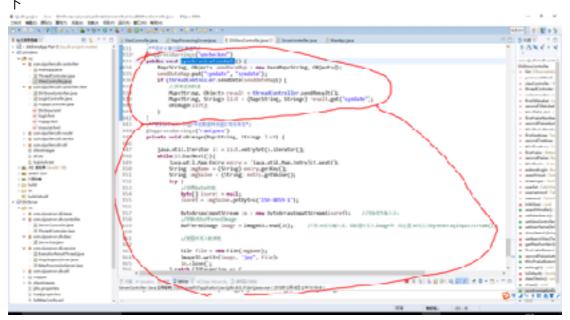
### 用户获取分享信息

## 前端

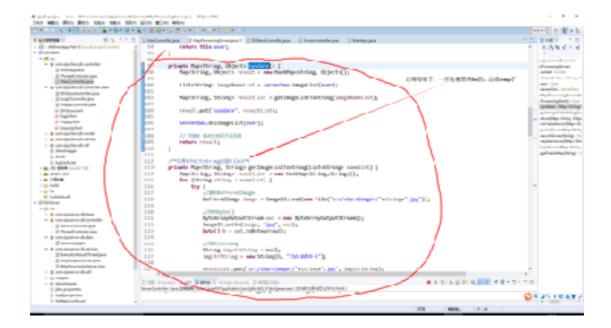
在加载页面的时候先把服务端的图片信息加载完,然后再出现界面

```
# Section of the Control of the Cont
```

服务端发来的是一个map,键是图片名,值是图片信息,这块注释写错了,我到时候改一



后端



# 打开本地文件夹

#### 判断一下操作系统,然后不同操作系统执行不同的自己的运行时shell就行了

```
### State | St
```

```
### SETTING | The Company of the Com
```