APP开发文档

**一．总体概述**

1.项目名称

富智

2.项目开发者（排名不分先后，按姓氏首字母顺序）

成员一：杜锡基

成员二：黄塨铭

成员三：廖启俊

成员四：

1. 项目开发环境

Android Studio 2.2 + Git

4.系统功能设定

移动端：

1. 可以实时查看各个传感器返回的数据。
2. 可以人工干预光照、空气湿度等。
3. 可以通过摄像头实时监控生产基地的情况。

硬件系统：

1. 负责采集各个传感器的数据并发送到服务器。
2. 接收服务器的控制指令去控制开关。

后台管理系统：

1. 接收硬件系统传送的指标数据，并对数据进行处理。
2. 接收移动端的数据请求，将数据（json）发送到移动端。
3. 接收移动端的控制请求，将指令（mqtt）发送到硬件系统。
4. 项目开发技术

Java

1. 设计思路

通过http协议实现对后台数据的获取，通过mqtt协议实现对硬件系统的控制。

1. 项目背景

本着实时监测、智能控制、专家方案的三大理念，为客户提供最优质的的服务项目，从多角度考虑需求，以求达到客户所需要的功能，实现零距离的操作。

**二．主要模块讲解**

模块一

1.名称：AutoLogin

2.简介：在已存在账号密码的情况下登录app

3.重要代码：

3.1功能：查看本地config中是否存在用户名来判断是否已经登录过

public boolean hasLogin() {

return !getUsername().equals("");

}

3.2 功能：检测到已经登录过后，发送网络请求去服务器请求登录后的token

private void getLoginObservable(final String username,

final String password){

//RXjava封装回调

}

3.3功能：登录成功跳转到主Activity，并将token保存到本地Common类

if(login != null){

Common.setToken(login.get("token"));

Common.setUsername(username);

Intent intent = new Intent(AutoLogin.this,

MainActivityNew.class);

startActivity(intent);

finish();

}

3.4功能：登录失败跳转到LoginActivity

ToastUtil.showShort(AutoLogin.this, "登陆失败，请重新登陆");

Intent intent = new Intent(AutoLogin.this, Login.class);

startActivity(intent);

finish();

模块二

1.名称：Login

2.简介：登录账户的主界面

3.重要代码：

3.1功能：检验用户名和密码的输入正确性（数字和字母）

private boolean checkUsernameAndPwd() {

Pattern pattern = Pattern.compile("^[0-9A-z]{3,30}$");

String strUsername = username.getText().toString();

Matcher usernameMatcher = pattern.matcher(strUsername);

if (!usernameMatcher.matches()) {

ToastUtil.showShort(this, "用户名格式不正确");

return false;

}

String strPwd = password.getText().toString();

Matcher pwdMatcher = pattern.matcher(strPwd);

if (!pwdMatcher.matches()) {

ToastUtil.showShort(this, "密码格式不正确");

return false;

}

return true;

}

3.2功能：点击登录按钮后发送网络请求去服务器请求登录后的token，并保存到Common和Config类

config.setToken(login.get("token"));

config.setUsername(username);

config.setPassword(password);

config.setLastUsername(username);

Common.setToken(login.get("token"));

Common.setUsername(username);

Intent intent = new Intent(Login.this, MainActivityNew.class);

startActivity(intent);

Login.this.finish();

3.3功能：点击忘记密码按钮，发送请求到服务器获取短信

findPwdHttp.request(phone, new FindPasswordHttp.MyListener(){

//接口回调

public void success(String find\_password\_status) {

//请求已经到达服务器，返回OK

}

public void loading() {

//等待服务器反馈信息

}

public void fail() {

//无法连接服务器

}

}

模块三

1.名称：IdentifyingCode

2.简介：注册界面的Activity

3.重要代码：

3.1功能：获取验证码

public void startConnect() {

getCodeHttp.request(phone, new MyListener() {

@Override

public void success() {

//验证码已经发到服务器

}

@Override

public void loading() {

//验证码正在发送

}

@Override

public void fail() {

progressDialog.dismiss();

//号码已经被注册（后续需要加接口判断）

}

});

}

3.2功能：倒计时类，负责刷新验证码按钮上的时间

public TimeCount(long millisInFuture, long countDownInterval) {

super(millisInFuture, countDownInterval);// 参数依次为总时长,和计时的时间间隔

}

@Override

public void onFinish() {// 计时完毕时触发

getYz.setText("重新验证");

getYz.setClickable(true);

}

@Override

public void onTick(long millisUntilFinished) {// 计时过程显示

getYz.setClickable(false);

getYz.setText("请等待" + millisUntilFinished / 1000 + "秒");

}

}

3.3功能：发送网络请求，验证手机号码与验证码是否匹配

private void checkValidCode(String phoneNum, String validCode) {

final List<BasicNameValuePair> params = new ArrayList<BasicNameValuePair>();

String jsonStr = String.format("{\"phone\":\"%s\", \"code\":\"%s\"}",

phoneNum, validCode);

params.add(new BasicNameValuePair("data", jsonStr));

Thread thread = new Thread(new Runnable() {

@Override

public void run() {

try {

showProgress();

String response = HttpUtil.httpPost(

new DefaultHttpClient(), uriAPI, params, "");

if (response == null) {

dismissProgress();

showValidFail();

} else {

JSONObject responseJson = new JSONObject(response);

String responseCode = (String) responseJson

.get("errmsg");

Log.d("gzfuzhi", responseCode);

if (responseCode.equals("ok")) {

dismissProgress();

showValidOk();

Config config = new Config(getApplication());

config.setUsername(phone);

switchToRegister(); // 跳转到注册界面

} else {

dismissProgress();

showValidFail();

}

}

} catch (Exception exception) {

Log.d("gzfuzhi", exception.getMessage());

dismissProgress();

showValidFail();

}

}

});

thread.start();

}

模块四

1.名称：EquipmentListFragment

2.简介：显示设备列表的fragment

3.重要代码：

3.1功能：搜索按钮的textChange事件

protected class SearchListener implements TextWatcher {

@Override

public void beforeTextChanged(CharSequence s, int start, int count, int after) {

}

//s改变后的String

@Override

public void onTextChanged(CharSequence s, int start, int before, int count) {

search\_layout\_background.setVisibility(View.GONE);

if (s.length() > 0) {

List<EquipmentBean> equipmentBeanList = new ArrayList<>();

for (int i = 0; i < equipmentBeans.size(); ++i) {

String equipmentCode = equipmentBeans.get(i).code;

if (Equipment.getEquipmentName(activity, equipmentCode).contains(s) || equipmentCode.contains(s)) {

equipmentBeanList.add(equipmentBeans.get(i));

}

}

searchListAdapter.setEquipmentBeanList(equipmentBeanList);

}

}

@Override

public void afterTextChanged(Editable s) {

}

}

3.2功能：listitem中的设置功能（根据是否置顶加载不同的menu）

public void onEvent(final EquipmentItemConfigureClickedEvent event) {

final PopupMenu equipmentConfigMenu = new PopupMenu(activity, event.view);

final Group group=new Group(activity,equipmentBeans);

TopBean topBean=group.getTopBean();

final String code=event.equipmentCode;

if(topBean.top.contains(code))

{

//加载有取消置顶功能的menu布局

}

else

{

//加载有置顶功能的menu布局

}

equipmentConfigMenu.show();

}

3.3功能：设置分组（核心代码）

SimpleAdapter adapter=new SimpleAdapter(

activity,

data,

R.layout.add\_to\_group\_dialog\_list\_item,

new String[]{"img","group"},

new int[]{R.id.add\_to\_group\_dialog\_img,R.id.add\_to\_group\_dialog\_title});

lstView.setAdapter(adapter);

lstView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

try {

new Group(activity, equipmentBeans).addChildToGroup(equipmentCode, modelGroup.getGroupNames().get(position));

} catch (Exception e) {

e.printStackTrace();

ToastUtil.showLong(activity, "设置分组失败。");

}

reloadView();

baseDialog.dismiss();

}

});

3.4功能：删除分组（核心代码）

positiveBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

try {

new Group(activity, equipmentBeans).deleteGroup(groupName);

} catch (Exception e) {

e.printStackTrace();

ToastUtil.showLong(activity, "删除分组失败。");

}

reloadView();

baseDialog.dismiss();

}

});

3.5功能：扫描二维码添加设备

protected void startBarcodeScanActivity() {

//调用Zxing扫码框架

Intent intent = new Intent(activity, com.zxing.activity.CaptureActivity.class);

startActivityForResult(intent, GET\_BARCODE);

}

//接受结果

public void onActivityResult(int requestCode, int resultCode, Intent data) {

if (resultCode != activity.RESULT\_OK)

return;

switch (requestCode) {

case GET\_BARCODE:

//扫二维码，获取Zxing扫码框架返回的结果，并上存到服务器

submitBarcode(data.getExtras().getString("result"));

break;

default:

break;

}

}

模块五

1. 名称：OverviewFragment
2. 简介：

3.重要代码：

模块六

1. 名称：ChartFragment
2. 简介：

3.重要代码：

模块七

1. 名称：SchemeDefaultFragment
2. 简介：

3.重要代码：

模块八

1. 名称：UserCenterFragment
2. 简介：

3.重要代码：