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，Android

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：主界面
3. 重要代码：

3.1视频：

protected ImageView realTimeBtn;

实现方法：  
protected class OnRealTimeClickedListener implements OnClickListener {  
 @Override  
 public void onClick(View v) {  
 EZOpenSDK.getInstance().openLoginPage();  
 }  
}

3.2拍照，换用户喜欢的照片：

protected ImageView userPhoto;

实现方法

protected class OnCameraButtonClickedListener implements OnClickListener {

@Override

public void onClick(View v) {

showTakePictureDialog();

}

}

3.3在主界面中显示的各种指标，湿度，光强，温度，二氧化碳等等  
 protected TableLayout buttonGroup;

protected void refreshView(List<RealtimeDataBean> beans) {

Log.d("gzfuzhi", "Start refresh view.");

// 刷新设备名字

loadEquipmentName();

// 刷新设备图像

loadEquipmentImage();

// 刷新实时数据

buttonGroup.removeAllViews();

//传感器事件

OnClickListener listener = new OnSensorClickedListener();

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

if (beans.get(i).getType().equals("GPS")){

beans.remove(i);

}

}

//行

TableRow row = null;

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

if (i % 3 == 0) {

row = new TableRow(activity);

LayoutParams param = new LayoutParams(

LayoutParams.MATCH\_PARENT, LayoutParams.WRAP\_CONTENT);

row.setLayoutParams(param);

}

View overviewItem = activity.getLayoutInflater().inflate(

R.layout.overview\_item, row, false);

ImageView iconIV = (ImageView) overviewItem

.findViewById(R.id.overview\_btn\_icon);

TextView valueNameTV = (TextView) overviewItem

.findViewById(R.id.overview\_value\_name);

TextView valueTV = (TextView) overviewItem

.findViewById(R.id.overview\_value);

TextView valueUnitTV = (TextView) overviewItem

.findViewById(R.id.overview\_value\_unit);

TextView valueDate = (TextView) overviewItem

.findViewById(R.id.overview\_value\_date);

RealtimeDataBean bean = beans.get(i);

String name = bean.getName();

String value = bean.getData().get(0);

String unit = bean.getUnit();

String type = bean.getType();

String updateDate = bean.getTime().get(0);

try {

Date d = DateUtil.dateFormat.parse(updateDate);

SimpleDateFormat format = new SimpleDateFormat("yy-MM-dd HH:mm", Locale.CHINA);

updateDate = format.format(d);

} catch (ParseException e) {

e.printStackTrace();

updateDate = "";

}

if (bean.getIsAlarm().get(0) == 1) {

valueTV.setTextColor(getResources().getColor(R.color.red\_2));

} else {

valueTV.setTextColor(

getResources().getColor(R.color.green\_2));

}

valueNameTV.setText(cutSensorName(name));

valueTV.setText(value);

valueUnitTV.setText(unit);

setIconWithType(iconIV, type);

valueDate.setText(updateDate);

row.addView(overviewItem);

overviewItem.setTag(i);

//画图事件，温度，光强，湿度等等

overviewItem.setOnClickListener(listener);

if (i % 3 == 0) {

buttonGroup.addView(row);

}

}

//占位符，这里好似有点问题，多了占位符，删去好似也没什么影响

for (int i = 0; i < beans.size() % 3; ++i) {

View overviewItem = activity.getLayoutInflater().inflate(

R.layout.overview\_item\_null, buttonGroup, false);

row.addView(overviewItem);

}

//历史数据需要获取sensorcode和time

//获取名字，湿度，光强，温度等等

ArrayList<String> names = new ArrayList<>();

for (int z = 0; z < beans.size(); ++z) {

names.add(cutSensorName(beans.get(z).getName()));

}

GlobalEvent.bus.post(new OverviewSensorCodeEvent(

names, selectedEquipmentCode,

getSensorCode(beans),getNewTime(beans)));

}

3.4下拉刷新:

protected SwipeRefreshLayout swipeRefreshLayout;

protected class PullToRefreshListener implements OnRefreshListener {

@Override

public void onRefresh() {

if (selectedEquipmentCode != null) {

RealtimeDataObservable.createObservable(Common.token,

selectedEquipmentCode)

.subscribeOn(Schedulers.newThread())

.subscribe(new ProgressSubscriber(new OnNextListener() {

@Override

public void onNext(Object obj) {

refreshView((List<RealtimeDataBean>) obj);

swipeRefreshLayout.setRefreshing(false);

}

}));

}

}

}

3.5用EventBus来传递事件,转到历史数据的界面：

在OverviewFragment中post

GlobalEvent.bus.post(new OverviewSensorCodeEvent(names,

selectedEquipmentCode,getSensorCode(beans),

getNewTime(beans));

在ChartFragment中注册处理

private class EventHandler{

public void onEvent(OverviewSensorCodeEvent overviewSensorCodeEvent){

sensorName = overviewSensorCodeEvent.sensorName;

equipmentCode = overviewSensorCodeEvent.equipmentCode;

sensorCode = overviewSensorCodeEvent.sensorCode;

historyDate = overviewSensorCodeEvent.newTime;

}

}

模块六

1. 名称：ChartFragment
2. 简介：历史数据
3. 重要代码：

3.1左右滑动显示不同传感器的历史数据

历史数据左右滑使用ViewPager实现：

private ViewPager chartViewPager;

ViewPager的每个view：ChartFragmentItem

ChartFragmentItem是用来画图的

chartViewPager.addOnPageChangeListener(new OnPageChangeListener() {

//此方法是页面跳转完后得到调用，arg0是你当前选中的页面的Position @Override

public void onPageSelected(int which) {

loadData(which);

}

//当页面在滑动的时候会调用此方法，在滑动被停止之前，此方法回一直 得到调用。

//arg0 :当前页面，及你点击滑动的页面

//arg1:当前页面偏移的百分比

//arg2:当前页面偏移的像素位置

@Override

public void onPageScrolled(int arg0, float arg1, int arg2) {

}

//此方法是在状态改变的时候调用，其中arg0这个参数有三种 状态（0，1，2）。

// arg0 ==1的时表示正在滑动，

// arg0==2的时表示滑动完毕了，

// arg0==0的时表示什么都没做。

@Override

public void onPageScrollStateChanged(int arg0) {

}

});

模块七

1. 名称：SchemeDefaultFragment
2. 简介：设备控制
3. 重要代码：

3.1控制器列表：

例如：喷淋，光强比，光强，湿度等控制

private TableLayout schemeDefaultIndicatorLayout;

3.2光强比的布局，进入控制设备默认显示的布局

private View lqcLayout;

3.3其他的布局

private LinearLayout othersParamLayout;

3.4喷淋功能

实现喷淋功能是使用mqtt协议，不经过服务器，手机直接和上位机传递信息。

主要通过两个class来实现WaterRoom和PublishWC来实现

浇水控制器信息

private static final String WATERCONTROLLER = "/c/shc/1";

3.5施肥功能（华南植物园新增功能）具体原理与喷淋一样。

施肥控制器信息

private static final String FERTILIZECONTROLLER = "/c/fc/1";

3.6光强，温度，湿度，二氧化碳等控制，这些控制与喷淋和施肥不一样，是 要上存服务器的。

指标值：指标值在上限与下限之间

上限：大于指标值

下限：小于指标值

上存：uploadScheme

private void uploadScheme(String indicatorKey) {  
 //为target、upper、lower填充参数  
 if (fillingParams(indicatorKey)) {  
 //选择人为干预的时间  
 fillingTime();  
 } }

3.7自定义光强

private void customLed() {

View contentView = View.inflate(activity, R.layout.fragment\_scheme\_new\_custom\_lqc, null);

final NumberPickerView ledRed = (NumberPickerView) contentView.findViewById(R.id.scheme\_custom\_led\_red);

final NumberPickerView ledBlue = (NumberPickerView) contentView.findViewById(R.id.scheme\_custom\_led\_blue);

final NumberPickerView ledWhite = (NumberPickerView) contentView.findViewById(R.id.scheme\_custom\_led\_white);

final BaseCustomAlterDialog baseDialog = new BaseCustomAlterDialog(activity);

// 0.65

RelativeLayout.LayoutParams contentLp = new RelativeLayout.LayoutParams(ViewGroup.LayoutParams.MATCH\_PARENT, ViewGroup.LayoutParams.WRAP\_CONTENT);

contentLp.addRule(RelativeLayout.CENTER\_IN\_PARENT, RelativeLayout.TRUE);

baseDialog.setWidthAndHeightRadio(0.8f, 0f)

.setTitle("自定义光强比")

.setNegativeBtnListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

baseDialog.dismiss();

}

})

.setPositiveBtnListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

lqc = ledRed.getValue() + ":" + ledBlue.getValue() + ":" + ledWhite.getValue();

uploadScheme("lqc");

baseDialog.dismiss();

}

})

.setContentView(contentView, contentLp);

}

3.8选择干预时间

private void fillingTime() {

String[] items = new String[]{"1分钟", "3分钟", "5分钟",

"10分钟", "自定义"};

startTime = "";

endTime = "";

//获取摆放到dialog中的listView

View contentView = View.inflate(activity, R.layout.dialog\_water\_choose\_time, null);

ListView listTime = (ListView) contentView.findViewById(R.id.id\_dialog\_water\_choose\_time\_list);

//设置listView的adapter

final DialogTimeListAdapter adapter = new DialogTimeListAdapter(activity, items);

listTime.setAdapter(adapter);

//为listView的item设置点击时换图片

listTime.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

//view为item复用的convertView

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

//先清空所有的图片

adapter.removeAllViews();

//再设置点击的item的图片变化

((ImageView) view.findViewById(

R.id.id\_water\_choose\_time\_list\_img))

.setImageResource(R.drawable.dialog\_img\_select);

//根据点击的listView的item剪辑startTime和endTime

countTime(position);

}

});

//设置dialog的基本属性，并将contentView加入到dialog中

final BaseCustomAlterDialog baseDialog = new BaseCustomAlterDialog(activity);

RelativeLayout.LayoutParams contentLp = new RelativeLayout.LayoutParams(ViewGroup.LayoutParams.MATCH\_PARENT, ViewGroup.LayoutParams.WRAP\_CONTENT);

contentLp.addRule(RelativeLayout.CENTER\_IN\_PARENT, RelativeLayout.TRUE);

// 0.55

baseDialog.setWidthAndHeightRadio(0.8f, 0f)

.setLocation(Gravity.CENTER, 0, 0)

.setNegativeBtnListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

baseDialog.dismiss();

}

})

.setPositiveBtnListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//检查target、upper、lower、startTime、endTime是否输正确，并上传

uploadAfterChecked();

baseDialog.dismiss();

}

})

.setTitle("请选择持续时间")

.setContentView(contentView, contentLp);

}

模块八

1. 名称：UserCenterFragment
2. 简介：我的
3. 重要代码：

3.1初始化设备：  
 private RelativeLayout deviceInit;

方法：

private void deviceInitMethod() {

Intent intent = new Intent(activity, WifiConfigActivity.class);

startActivity(intent);

}

3.2修改当前密码：

private RelativeLayout passwordChange;

方法：

private void passwordChangeMethod() {

Intent intent = new Intent(activity, Password.class);

activity.startActivity(intent);

}

3.3退出当前账号：

private RelativeLayout exit;

方法：

private void exitMethod() {

new Config(activity).clear();

Intent intent = new Intent(activity, Login.class);

activity.startActivity(intent);

activity.finish();

}

3.4关于：

private RelativeLayout about;

方法：

private void showAbout() {

View contentView = View.inflate(activity,R.layout.dialog\_about,null);

final BaseAlterDialogUtil baseDialog = new BaseAlterDialogUtil(activity);

TextView tvContent = (TextView) contentView.findViewById(

R.id.id\_dialog\_about\_content);

tvContent.setText(

"\u3000\u3000自然一号是富智精准智慧农业物联网采控终端。

\n" +"【实时检测】可实时采集空气或土壤的温湿度，空气CO2

浓度、光照强度、土壤/水质PH、土壤EC值、溶解氧等环境因子。\n" +

"【智能控制】可对智能LED植物照明、滴灌、喷淋、水泵、风机等设备进行

智能控制。\n" +

"【专家方案】系统设立专家帐户，专家设置的技术方案与设施管理方案可自 动匹配，智能运行。");

Button btnBack = (Button) contentView.findViewById(

R.id.id\_dialog\_about\_btn);

btnBack.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

baseDialog.dismiss();

}

});

baseDialog

.setLocation(Gravity.CENTER,0,0)

.setWidthAndHeightRadio(0.8f,0.65f)

.setContentView(contentView);

}

3.5检查更新：

private RelativeLayout update;

方法：

private void updateMethod() {

UpdateManager updateManager = UpdateManager

.getInstance(activity);

updateManager.checkUpdate(new CheckUpdateCallback() {

BaseProgressDialog progressDialog = new BaseProgressDialog(

activity);

@Override

public void onCheckUpdateFinished(final int currentVersion,

final int lastestVersion, final String url,

final String date, final String note) {

activity.runOnUiThread(new Runnable() {

@Override

public void run() {

progressDialog.dismiss();

AlertDialog.Builder builder =

new AlertDialog.Builder(

activity);

if (currentVersion == -1

|| lastestVersion == -1) {

if (currentVersion == -1) {

builder.setMessage("获取本地版本号出错");

} else {

builder.setMessage("获取远端版本号出错");

}

builder.setCancelable(false)

.setPositiveButton("好的",

new DialogInterface.OnClickListener() {

@Override

public void onClick(

DialogInterface dialog,

int which) {

dialog.cancel();

}

});

} else if (lastestVersion > currentVersion) {

builder.setTitle("更新日期: " + date)

.setMessage(note)

.setCancelable(false)

.setPositiveButton("更新",

new DialogInterface.OnClickListener() {

public void onClick(

DialogInterface dialog,

int id) {

UpdateManager.getInstance(

activity).downloadAndInstall(url);

}

})

.setNegativeButton(

"下一次",

new DialogInterface.OnClickListener() {

public void onClick(

DialogInterface dialog, int id) {

dialog.cancel();

}

});

} else {

builder.setMessage("已经是最新版了")

.setCancelable(true)

.setPositiveButton(

"好的",

new DialogInterface.OnClickListener() {

@Override

public void onClick(

DialogInterface dialog,

int which) {

dialog.cancel();

}

});

}

AlertDialog alert = builder.create();

if (!activity.isFinishing())

alert.show();

}

});

}