**public class** EquipmentActivity **extends** BaseActivity {  
  
 @BindView(R.id.***image\_back***)  
 ImageView **imageBack**;  
 @BindView(R.id.***text\_top***)  
 TextView **textTop**;  
 @BindView(R.id.***image\_add***)  
 ImageView **imageAdd**;  
 @BindView(R.id.***pull\_icon***)  
 ImageView **pullIcon**;  
 @BindView(R.id.***refreshing\_icon***)  
 ImageView **refreshingIcon**;  
 @BindView(R.id.***state\_tv***)  
 TextView **stateTv**;  
 @BindView(R.id.***state\_iv***)  
 ImageView **stateIv**;  
 @BindView(R.id.***head\_view***)  
 RelativeLayout **headView**;  
 @BindView(R.id.***recycler\_message***)  
 RecyclerView **recyclerMessage**;  
 @BindView(R.id.***pullup\_icon***)  
 ImageView **pullupIcon**;  
 @BindView(R.id.***loading\_icon***)  
 ImageView **loadingIcon**;  
 @BindView(R.id.***loadstate\_tv***)  
 TextView **loadstateTv**;  
 @BindView(R.id.***loadstate\_iv***)  
 ImageView **loadstateIv**;  
 @BindView(R.id.***loadmore\_view***)  
 RelativeLayout **loadmoreView**;  
 @BindView(R.id.***refresh\_view***)  
 PullToRefreshLayout **refreshView**;  
 @BindView(R.id.***linear\_no\_data***)  
 LinearLayout **linearNoData**;  
 @BindView(R.id.***edit\_user\_search***)  
 EditText **editUserSearch**;  
 @BindView(R.id.***image\_user\_search***)  
 ImageView **imageUserSearch**;  
 @BindView(R.id.***text\_user\_search***)  
 TextView **textUserSearch**;  
  
 **private int page** = 1;  
 **private** ArrayList<UserModel> **arrayList**;  
 **private** EquipmentAdapter **adapter**;  
 **private** Dialog **dialog\_tips**;  
 **private** PopupWindow **popupWindow**;  
 **private** ArrayList<UserModel> **string**;  
 **private** String[] **strings**;  
 **private int position**;  
 **private** String **id**, **sid** = **""**;  
 **private** String **number\_search** = **""**, **type\_search** = **""**;  
 **private** View **layout\_type**;  
 **private** String **status**;  
 **private int type** = 1;  
 **private int REQUEST\_CODE\_SCAN** = 100;  
  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***ui\_user***);  
 ButterKnife.*bind*(**this**);  
 **imageBack**.setOnClickListener(v -> finish());  
 **imageAdd**.setVisibility(View.***VISIBLE***);  
 **textTop**.setText(**"设备管理"**);  
 **textUserSearch**.setText(**"全部设备"**);  
 **editUserSearch**.setHint(**"搜索设备编号"**);  
 *//设置固定大小* **recyclerMessage**.setHasFixedSize(**true**);  
 *//创建线性布局* LinearLayoutManager mLayoutManager = **new** LinearLayoutManager(**this**);  
 *//垂直方向* mLayoutManager.setOrientation(OrientationHelper.***VERTICAL***);  
 *//给RecyclerView设置布局管理器* **recyclerMessage**.setLayoutManager(mLayoutManager);  
 **arrayList** = **new** ArrayList<>();  
 **adapter** = **new** EquipmentAdapter(**this**, **arrayList**, **clickListener**);  
 **recyclerMessage**.setAdapter(**adapter**);  
  
 **refreshView**.setOnRefreshListener(**new** PullToRefreshLayout.OnRefreshListener() {  
 @Override  
 **public void** onRefresh(PullToRefreshLayout pullToRefreshLayout) {  
 **page** = 1;  
 *// 下拉刷新操作* **arrayList**.clear();  
 **try** {  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 **public void** onLoadMore(PullToRefreshLayout pullToRefreshLayout) {  
 **page**++;  
 **try** {  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 **imageAdd**.setOnClickListener(v -> dialog\_Add());  
 **try** {  
 **blockDialog**.show();  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **textUserSearch**.setOnClickListener(v -> {  
 **string** = **new** ArrayList<>();  
 **type** = 7;  
 **strings** = **new** String[]{**"全部设备"**, **"绑定人员"**, **"未绑定人员"**};  
 **for** (**int** i = 0; i < **strings**.**length**; i++) {  
 UserModel model = **new** UserModel();  
 model.setTitle(**strings**[i]);  
 model.setId(i + **""**);  
 **string**.add(model);  
 }  
 popwindow\_keyword(**textUserSearch**, **type**);  
 });  
 **imageUserSearch**.setOnClickListener(v -> {  
 **number\_search** = **editUserSearch**.getText().toString();  
 **if** (!**number\_search**.equals(**""**)) {  
 **arrayList**.clear();  
 **page** = 1;  
 **try** {  
 **blockDialog**.show();  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
  
 **private void** postLogin() **throws** JSONException {  
 JSONObject json = **new** JSONObject();  
 json.put(**"uid"**, Share.*getUid*(**this**));  
 HttpUtils.*getInstance*().postJson(ConfigNet.***GET\_USER\_TYPE***, json.toString(), ConfigNet.***GET\_USER\_TYPE\_CODE***, Share.*getToken*(**this**), **handler**);  
 }  
  
 **private void** postList() **throws** JSONException {  
 JSONObject json = **new** JSONObject();  
 json.put(**"page"**, **page**);  
 json.put(**"limit"**, **"10"**);  
 json.put(**"number"**, **number\_search**);  
 json.put(**"uid"**, Share.*getUid*(**this**));  
 json.put(**"type"**, **type\_search**);  
 HttpUtils.*getInstance*().postJson(ConfigNet.***GET\_EQUIPMENT\_LIST***, json.toString(), ConfigNet.***GET\_EQUIPMENT\_LIST\_CODE***, Share.*getToken*(**this**), **handler**);  
 }  
  
 **private void** postJie() **throws** JSONException {  
 JSONObject json = **new** JSONObject();  
 json.put(**"id"**, **id**);  
 json.put(**"uid"**, Share.*getUid*(**this**));  
 HttpUtils.*getInstance*().postJson(ConfigNet.***GET\_USER\_JIE***, json.toString(), ConfigNet.***GET\_USER\_JIE\_CODE***, Share.*getToken*(**this**), **handler**);  
 }  
  
 **private void** postAdd() **throws** JSONException {  
 JSONObject json = **new** JSONObject();  
 json.put(**"number"**, **number**);  
 json.put(**"hats"**, **hats**);  
 json.put(**"name"**, **name**);  
 json.put(**"mobile"**, **mobile**);  
 json.put(**"category"**, **category**);  
 json.put(**"uid"**, Share.*getUid*(**this**));  
 json.put(**"sex"**, **sex**);  
 HttpUtils.*getInstance*().postJson(ConfigNet.***GET\_EQUIPMENT\_ADD***, json.toString(), ConfigNet.***GET\_EQUIPMENT\_ADD\_CODE***, Share.*getToken*(**this**), **handler**);  
 }  
  
 **private void** postBang() **throws** JSONException {  
 JSONObject json = **new** JSONObject();  
 json.put(**"name"**, **name**);  
 json.put(**"mobile"**, **mobile**);  
 json.put(**"category"**, **category**);  
 json.put(**"uid"**, Share.*getUid*(**this**));  
 json.put(**"sid"**, **id**);  
 HttpUtils.*getInstance*().postJson(ConfigNet.***GET\_EQUIPMENT\_BANG***, json.toString(), ConfigNet.***GET\_EQUIPMENT\_BANG\_CODE***, Share.*getToken*(**this**), **handler**);  
 }  
  
 **private void** postChange() **throws** JSONException {  
 JSONObject json = **new** JSONObject();  
 json.put(**"id"**, **id**);  
 json.put(**"uid"**, Share.*getUid*(**this**));  
 json.put(**"status"**, **status**);  
 HttpUtils.*getInstance*().postJson(ConfigNet.***GET\_EQUIPMENT\_CHANGE***, json.toString(), ConfigNet.***GET\_EQUIPMENT\_CHANGE\_CODE***, Share.*getToken*(**this**), **handler**);  
 }  
  
 **private void** postPhone(String ids,String uid) **throws** JSONException {  
 JSONObject json = **new** JSONObject();  
 json.put(**"id"**, ids);  
 json.put(**"uid"**, uid);  
 json.put(**"phone"**, SYApplication.*getAndroidId*(**this**));  
 HttpUtils.*getInstance*().postJson(ConfigNet.***GET\_PHONE\_BANG***, json.toString(), ConfigNet.***GET\_PHONE\_BANG\_CODE***, Share.*getToken*(**this**), **handler**);  
 }  
  
 *//* @SuppressLint(**"HandlerLeak"**)  
 Handler **handler** = **new** Handler() {  
 @Override  
 **public void** handleMessage(Message msg) {  
 **super**.handleMessage(msg);  
 String str = msg.**obj**.toString();  
 **blockDialog**.dismiss();  
 **switch** (msg.**what**) {  
 **case** ConfigNet.***GET\_USER\_TYPE\_CODE***:  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(str);  
 **int** code = jsonObject.getInt(**"code"**);  
 String message = jsonObject.getString(**"msg"**);  
 **if** (code == 200) {  
 String data = jsonObject.getString(**"data"**);  
 **if** (!data.equals(**"null"**)) {  
 ArrayList<UserModel> resultArr = **gson**.fromJson(data, **new** TypeToken<ArrayList<UserModel>>() {  
 }.getType());  
 **string** = **new** ArrayList<>();  
 **string**.addAll(resultArr);  
 popwindow\_keyword(**layout\_type**, **type**);  
 }  
  
 } **else** {  
 ToastUtil.*show*(message);  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **break**;  
 **case** ConfigNet.***GET\_EQUIPMENT\_LIST\_CODE***:  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(str);  
 **int** code = jsonObject.getInt(**"code"**);  
 **refreshView**.loadmoreFinish(PullToRefreshLayout.***SUCCEED***);  
 **refreshView**.refreshFinish(PullToRefreshLayout.***SUCCEED***);  
 **if** (code == 200) {  
 **refreshView**.setVisibility(View.***VISIBLE***);  
 **linearNoData**.setVisibility(View.***GONE***);  
 String data = jsonObject.getString(**"data"**);  
 **if** (!data.equals(**"null"**)) {  
 ArrayList<UserModel> resultArr = **gson**.fromJson(data, **new** TypeToken<ArrayList<UserModel>>() {  
 }.getType());  
 **arrayList**.addAll(resultArr);  
 **adapter**.notifyDataSetChanged();  
 } **else** {  
 **if** (**page** == 1) {  
 **refreshView**.setVisibility(View.***GONE***);  
 **linearNoData**.setVisibility(View.***VISIBLE***);  
 }  
 }  
  
 } **else** {  
 **if** (**page** == 1) {  
 **refreshView**.setVisibility(View.***GONE***);  
 **linearNoData**.setVisibility(View.***VISIBLE***);  
 }  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **break**;  
 **case** ConfigNet.***GET\_EQUIPMENT\_ADD\_CODE***:  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(str);  
 **int** code = jsonObject.getInt(**"code"**);  
 String message = jsonObject.getString(**"msg"**);  
 **if** (code == 200) {  
 **arrayList**.clear();  
 **try** {  
 **blockDialog**.show();  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **dialog\_tips**.dismiss();  
 ToastUtil.*show*(message);  
 } **else** {  
 ToastUtil.*show*(message);  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **break**;  
 **case** ConfigNet.***GET\_EQUIPMENT\_BANG\_CODE***:  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(str);  
 **int** code = jsonObject.getInt(**"code"**);  
 String message = jsonObject.getString(**"msg"**);  
 **if** (code == 200) {  
 **arrayList**.clear();  
 **try** {  
 **blockDialog**.show();  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **dialog\_tips**.dismiss();  
 ToastUtil.*show*(message);  
 } **else** {  
 ToastUtil.*show*(message);  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **break**;  
 **case** ConfigNet.***GET\_EQUIPMENT\_CHANGE\_CODE***:  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(str);  
 **int** code = jsonObject.getInt(**"code"**);  
 String message = jsonObject.getString(**"msg"**);  
 **if** (code == 200) {  
 **if** (**status**.equals(**"1"**)) {  
 **arrayList**.get(**position**).setStatus(**"1"**);  
 **adapter**.notifyItemChanged(**position**);  
 } **else if** (**status**.equals(**"2"**)) {  
 **arrayList**.get(**position**).setStatus(**"2"**);  
 **adapter**.notifyItemChanged(**position**);  
 } **else if** (**status**.equals(**"3"**)) {  
 **arrayList**.remove(**position**);  
 **adapter**.notifyDataSetChanged();  
 }  
 } **else** {  
 ToastUtil.*show*(message);  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **break**;  
 **case** ConfigNet.***GET\_USER\_JIE\_CODE***:  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(str);  
 **int** code = jsonObject.getInt(**"code"**);  
 String message = jsonObject.getString(**"msg"**);  
 **if** (code == 200) {  
 **arrayList**.clear();  
 **try** {  
 **blockDialog**.show();  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 } **else** {  
 ToastUtil.*show*(message);  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **break**;  
 **case** ConfigNet.***GET\_PHONE\_BANG\_CODE***:  
 Log.*i*(**"toby"**, **"handleMessage: "** + str);  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(str);  
 **int** code = jsonObject.getInt(**"code"**);  
 String message = jsonObject.getString(**"msg"**);  
 **if** (code == 200) {  
 ToastUtil.*show*(message);  
 } **else** {  
 ToastUtil.*show*(message);  
 }  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **break**;  
 }  
 }  
 };  
  
 **private** View.OnClickListener **clickListener** = v -> {  
 **switch** (v.getId()) {  
 **case** R.id.***text\_user\_zhu***:*// 注销账户* **position** = (Integer) v.getTag();  
 **status** = **"3"**;  
 **id** = **arrayList**.get(**position**).getId();  
 **if** (**arrayList**.get(**position**).getUser() == **null**) {  
 dialog\_zhu(**arrayList**.get(**position**).getNumber(), **""**,  
 **""**, **arrayList**.get(**position**).getStatus());  
 } **else** {  
 dialog\_zhu(**arrayList**.get(**position**).getNumber(), **arrayList**.get(**position**).getUser().getNameX(),  
 **arrayList**.get(**position**).getUser().getCategoryX(), **arrayList**.get(**position**).getStatus());  
 }  
 **break**;  
 **case** R.id.***text\_user\_dong***:*// 解冻/冻结账号* **position** = (Integer) v.getTag();  
 **id** = **arrayList**.get(**position**).getId();  
 **if** (**arrayList**.get(**position**).getStatus().equals(**"1"**)) {  
 **status** = **"2"**;  
 dialog\_jie(1);  
 } **else** {  
 **status** = **"1"**;  
 dialog\_jie(3);  
 }  
  
 **break**;  
 **case** R.id.***text\_user\_jie***:*// 绑定/解绑设备* **position** = (Integer) v.getTag();  
*// if (arrayList.get(position).getUser() == null) {  
// id = arrayList.get(position).getId();  
// dialog\_bang(arrayList.get(position).getNumber(), arrayList.get(position).getHats());  
// } else {  
// id = arrayList.get(position).getUser().getIdX();  
// dialog\_jie(2);  
// }* AndPermission.*with*(**this**).permission(Permission.***CAMERA***, Permission.***READ\_EXTERNAL\_STORAGE***)  
 .onGranted(permissions -> {  
 Intent intent = **new** Intent(EquipmentActivity.**this**, CaptureActivity.**class**);  
 */\*ZxingConfig是配置类  
 \*可以设置是否显示底部布局，闪光灯，相册，  
 \* 是否播放提示音 震动  
 \* 设置扫描框颜色等  
 \* 也可以不传这个参数  
 \* \*/* ZxingConfig config = **new** ZxingConfig();  
 config.setPlayBeep(**true**);*//是否播放扫描声音 默认为true* config.setShake(**true**);*//是否震动 默认为true* config.setDecodeBarCode(**false**);*//是否扫描条形码 默认为true* config.setReactColor(R.color.***white***);*//设置扫描框四个角的颜色 默认为淡蓝色* config.setFrameLineColor(R.color.***white***);*//设置扫描框边框颜色 默认无色* config.setFullScreenScan(**true**);*//是否全屏扫描 默认为true 设为false则只会在扫描框中扫描* intent.putExtra(Constant.***INTENT\_ZXING\_CONFIG***, config);  
 startActivityForResult(intent, **REQUEST\_CODE\_SCAN**);  
 }).onDenied(permissions -> {  
 Uri packageURI = Uri.*parse*(**"package:"** + getPackageName());  
 Intent intent = **new** Intent(Settings.***ACTION\_APPLICATION\_DETAILS\_SETTINGS***, packageURI);  
 intent.addFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
  
 startActivity(intent);  
  
*// Toast.makeText(GoodsActivity.this, "没有权限无法扫描呦", Toast.LENGTH\_LONG).show();* }).start();  
 **break**;  
  
 }  
 };  
  
 **private** String **hats** = **""**, **number** = **""**, **name** = **""**, **mobile** = **""**, **sex** = **""**, **category** = **""**;  
  
 **private void** dialog\_Add() {  
 LayoutInflater inflater\_type = getLayoutInflater();  
 **layout\_type** = inflater\_type.inflate(R.layout.***dialog\_equipment\_add***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(**this**);  
 record\_type.setView(**new** EditText(**this**));  
 **dialog\_tips** = record\_type.create();  
 ButterKnife.*bind*(**layout\_type**);  
 **dialog\_tips**.getWindow().setBackgroundDrawableResource(android.R.color.***transparent***);*//设置Dialog背景透明效果* **dialog\_tips**.show();  
 **dialog\_tips**.setContentView(**layout\_type**);  
 **dialog\_tips**.setCanceledOnTouchOutside(**true**);*//* WindowManager windowManager = getWindowManager();  
 Display display = windowManager.getDefaultDisplay();  
 WindowManager.LayoutParams lp = **dialog\_tips**.getWindow()  
 .getAttributes();  
 lp.**width** = ((display.getWidth()) / 6) \* 5;  
 **dialog\_tips**.getWindow().setAttributes(lp);  
 addViewHolder viewHolder = **new** addViewHolder(**layout\_type**);  
 viewHolder.**btnAddSure**.setOnClickListener(v -> {  
 **number** = viewHolder.**editAddNumber**.getText().toString();  
 **name** = viewHolder.**editAddName**.getText().toString();  
 **mobile** = viewHolder.**editAddPhone**.getText().toString();  
 **sex** = viewHolder.**textAddSex**.getText().toString();  
 **category** = viewHolder.**textAddType**.getText().toString();  
 **if** (**number**.equals(**""**)) {  
 ToastUtil.*show*(**"设备号不能为空"**);  
 } **else** {  
 **if** (**name**.equals(**""**) && **mobile**.equals(**""**) && **sex**.equals(**""**) && **category**.equals(**""**)) {  
 **try** {  
 **blockDialog**.show();  
 postAdd();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 } **else** {  
 **try** {  
 **blockDialog**.show();  
 postAdd();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 });  
 viewHolder.**relativeAddSex**.setOnClickListener(v -> {  
 **if** (**popupWindow** != **null**)  
 **if** (**popupWindow**.isShowing()) {  
 **popupWindow**.dismiss();  
 }  
 **string** = **new** ArrayList<>();  
 **strings** = **new** String[]{**"男"**, **"女"**};  
 **for** (**int** i = 0; i < **strings**.**length**; i++) {  
 UserModel model = **new** UserModel();  
 model.setTitle(**strings**[i]);  
 **string**.add(model);  
 }  
 **type** = 1;  
 popwindow\_keyword(**layout\_type**, **type**);  
 });  
 viewHolder.**relativeAddType**.setOnClickListener(v -> {  
 **if** (**popupWindow** != **null**)  
 **if** (**popupWindow**.isShowing()) {  
 **popupWindow**.dismiss();  
 }  
 **type** = 2;  
 **try** {  
 **blockDialog**.show();  
 postLogin();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 });  
 viewHolder.**relativeAddColor**.setOnClickListener(v -> {  
 **if** (**popupWindow** != **null**)  
 **if** (**popupWindow**.isShowing()) {  
 **popupWindow**.dismiss();  
 }  
 *//1白色 2蓝色 3红色 4黄色* **string** = **new** ArrayList<>();  
 **strings** = **new** String[]{**"白色"**, **"蓝色"**, **"红色"**, **"黄色"**};  
 **for** (**int** i = 0; i < **strings**.**length**; i++) {  
 UserModel model = **new** UserModel();  
 model.setTitle(**strings**[i]);  
 model.setId((i + 1) + **""**);  
 **string**.add(model);  
 }  
 **type** = 5;  
 popwindow\_keyword(**layout\_type**, 5);  
 });  
  
 }  
  
 **private void** popwindow\_keyword(View key, **int** type) {  
 View popview = LayoutInflater.*from*(**this**).inflate(  
 R.layout.***dialog\_list***, **null**);  
 WindowManager windowManager = getWindowManager();  
 Display display = windowManager.getDefaultDisplay();  
 **int** width;  
 **if** (type == 1) {  
 width = (((display.getWidth()) / 6) \* 5) / 2;  
 } **else if** (type == 7) {  
 width = 400;  
 } **else** {  
 width = (((display.getWidth()) / 6) \* 5);  
 }  
 **popupWindow** = **new** PopupWindow(popview, width,  
 DrawerLayout.LayoutParams.***WRAP\_CONTENT***);  
 **popupWindow**.setFocusable(**false**);  
 **popupWindow**.setOutsideTouchable(**false**);  
 **popupWindow**.setBackgroundDrawable(**new** BitmapDrawable());  
 RecyclerView recycle\_city = popview  
 .findViewById(R.id.***recycler\_list***);  
 LinearLayoutManager linearLayoutManager;  
 DialogAdapter adapter = **new** DialogAdapter(**this**, **string**);  
 *//设置固定大小* recycle\_city.setHasFixedSize(**true**);  
 *//创建线性布局* linearLayoutManager = **new** LinearLayoutManager(**this**);  
 *//垂直方向* linearLayoutManager.setOrientation(OrientationHelper.***VERTICAL***);  
 *//给RecyclerView设置布局管理器* recycle\_city.setLayoutManager(linearLayoutManager);  
 recycle\_city.setAdapter(adapter);  
 adapter.notifyDataSetChanged();  
  
 adapter.setOnItemClickListener((view, position) -> {  
 **popupWindow**.dismiss();  
 **if** (type == 1) {  
 addViewHolder viewHolder = **new** addViewHolder(key);  
 viewHolder.**textAddSex**.setText(**string**.get(position).getTitle());  
 } **else if** (type == 2) {  
 addViewHolder viewHolder = **new** addViewHolder(key);  
 viewHolder.**textAddType**.setText(**string**.get(position).getTitle());  
 } **else if** (type == 4) {  
 bangViewHolder viewHolder = **new** bangViewHolder(key);  
 viewHolder.**textBangType**.setText(**string**.get(position).getTitle());  
 } **else if** (type == 5) {  
 addViewHolder viewHolder = **new** addViewHolder(key);  
 **hats** = **string**.get(position).getId();  
 viewHolder.**textAddColor**.setText(**string**.get(position).getTitle());  
 } **else if** (type == 7) {  
 **textUserSearch**.setText(**string**.get(position).getTitle());  
 **type\_search** = **string**.get(position).getId();  
 **if** (**type\_search**.equals(**"0"**)) {  
 **type\_search** = **""**;  
 }  
 **try** {  
 **arrayList**.clear();  
 **page** = 1;  
 **blockDialog**.show();  
 postList();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 **if** (**popupWindow**.isShowing()) {  
 **popupWindow**.dismiss();  
 } **else** {  
 **if** (type == 1) {  
 addViewHolder viewHolder = **new** addViewHolder(key);  
 **popupWindow**.showAsDropDown(viewHolder.**relativeAddSex**);  
 } **else if** (type == 2) {  
 addViewHolder viewHolder = **new** addViewHolder(key);  
 **popupWindow**.showAsDropDown(viewHolder.**relativeAddType**);  
 } **else if** (type == 4) {  
 bangViewHolder viewHolder = **new** bangViewHolder(key);  
 **popupWindow**.showAsDropDown(viewHolder.**relativeBangType**);  
 } **else if** (type == 5) {  
 addViewHolder viewHolder = **new** addViewHolder(key);  
 **popupWindow**.showAsDropDown(viewHolder.**relativeAddColor**);  
 } **else if** (type == 7) {  
 **popupWindow**.showAsDropDown(**textUserSearch**);  
 }  
 }  
 }  
  
 **private void** dialog\_bang(String number, String sex) {  
 LayoutInflater inflater\_type = getLayoutInflater();  
 **layout\_type** = inflater\_type.inflate(R.layout.***dialog\_bang\_person***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(**this**);  
 record\_type.setView(**new** EditText(**this**));  
 **dialog\_tips** = record\_type.create();  
 ButterKnife.*bind*(**layout\_type**);  
 **dialog\_tips**.getWindow().setBackgroundDrawableResource(android.R.color.***transparent***);*//设置Dialog背景透明效果* **dialog\_tips**.show();  
 **dialog\_tips**.setContentView(**layout\_type**);  
 **dialog\_tips**.setCanceledOnTouchOutside(**true**);*//* WindowManager windowManager = getWindowManager();  
 Display display = windowManager.getDefaultDisplay();  
 WindowManager.LayoutParams lp = **dialog\_tips**.getWindow()  
 .getAttributes();  
 lp.**width** = ((display.getWidth()) / 6) \* 5 - 80;  
 **dialog\_tips**.getWindow().setAttributes(lp);  
 bangViewHolder viewHolder = **new** bangViewHolder(**layout\_type**);  
 viewHolder.**editBangNumber**.setText(number);  
 *//"白色", "蓝色", "红色", "黄色"* **if** (sex.equals(**"1"**)) {  
 viewHolder.**textBangColor**.setText(**"白色"**);  
 } **else if** (sex.equals(**"2"**)) {  
 viewHolder.**textBangColor**.setText(**"蓝色"**);  
 } **else if** (sex.equals(**"3"**)) {  
 viewHolder.**textBangColor**.setText(**"红色"**);  
 } **else if** (sex.equals(**"4"**)) {  
 viewHolder.**textBangColor**.setText(**"黄色"**);  
 }  
  
 viewHolder.**btnAddSure**.setOnClickListener(v -> {  
 **name** = viewHolder.**editBangName**.getText().toString();  
 **mobile** = viewHolder.**editBangPhone**.getText().toString();  
 **category** = viewHolder.**textBangType**.getText().toString();  
 **if** (**name**.equals(**""**)) {  
 ToastUtil.*show*(**"人员姓名不能为空"**);  
 } **else** {  
 **try** {  
 **blockDialog**.show();  
 postBang();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 viewHolder.**relativeBangType**.setOnClickListener(v -> {  
 **if** (**popupWindow** != **null**)  
 **if** (**popupWindow**.isShowing()) {  
 **popupWindow**.dismiss();  
 }  
 **type** = 4;  
 **try** {  
 **blockDialog**.show();  
 postLogin();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 });  
 }  
  
 **private void** dialog\_zhu(String number, String name, String type, String status) {  
 LayoutInflater inflater\_type = getLayoutInflater();  
 View layout\_type = inflater\_type.inflate(R.layout.***dialog\_zhu***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(**this**);  
 **dialog\_tips** = record\_type.create();  
 ButterKnife.*bind*(layout\_type);  
 **dialog\_tips**.getWindow().setBackgroundDrawableResource(android.R.color.***transparent***);*//设置Dialog背景透明效果* **dialog\_tips**.show();  
 **dialog\_tips**.setContentView(layout\_type);  
 **dialog\_tips**.setCanceledOnTouchOutside(**true**);*//* WindowManager windowManager = getWindowManager();  
 Display display = windowManager.getDefaultDisplay();  
 WindowManager.LayoutParams lp = **dialog\_tips**.getWindow()  
 .getAttributes();  
 lp.**width** = ((display.getWidth()) / 6) \* 5;  
 **dialog\_tips**.getWindow().setAttributes(lp);  
 Button btn\_zhu\_sure = layout\_type.findViewById(R.id.***btn\_zhu\_sure***);  
 TextView text\_zhu\_status = layout\_type.findViewById(R.id.***text\_zhu\_status***);  
 TextView text\_zhu\_type = layout\_type.findViewById(R.id.***text\_zhu\_type***);  
 TextView text\_zhu\_number = layout\_type.findViewById(R.id.***text\_zhu\_number***);  
 TextView text\_zhu\_name = layout\_type.findViewById(R.id.***text\_zhu\_name***);  
 TextView text\_zhu\_title = layout\_type.findViewById(R.id.***text\_zhu\_title***);  
 text\_zhu\_title.setText(**"注销设备"**);  
 text\_zhu\_name.setText(number);  
 text\_zhu\_number.setText(name);  
 text\_zhu\_type.setText(type);  
 **if** (status.equals(**"1"**)) {  
 text\_zhu\_status.setText(**"正常状态"**);  
 } **else** {  
 text\_zhu\_status.setText(**"冻结状态"**);  
 }  
 btn\_zhu\_sure.setText(**"确认注销设备"**);  
 btn\_zhu\_sure.setOnClickListener(v -> {  
 **try** {  
 **blockDialog**.show();  
 postChange();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **dialog\_tips**.dismiss();  
 });  
  
 }  
  
 **private void** dialog\_jie(**int** type) {  
 LayoutInflater inflater\_type = getLayoutInflater();  
 View layout\_type = inflater\_type.inflate(R.layout.***dialog\_jie***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(**this**);  
 **dialog\_tips** = record\_type.create();  
 ButterKnife.*bind*(layout\_type);  
 **dialog\_tips**.getWindow().setBackgroundDrawableResource(android.R.color.***transparent***);*//设置Dialog背景透明效果* **dialog\_tips**.show();  
 **dialog\_tips**.setContentView(layout\_type);  
 **dialog\_tips**.setCanceledOnTouchOutside(**true**);*//* WindowManager windowManager = getWindowManager();  
 Display display = windowManager.getDefaultDisplay();  
 WindowManager.LayoutParams lp = **dialog\_tips**.getWindow()  
 .getAttributes();  
 lp.**width** = ((display.getWidth()) / 6) \* 5;  
 **dialog\_tips**.getWindow().setAttributes(lp);  
 Button btn\_jie\_sure = layout\_type.findViewById(R.id.***btn\_jie\_sure***);  
 Button btn\_jie\_cancel = layout\_type.findViewById(R.id.***btn\_jie\_cancel***);  
 TextView text\_jie\_content = layout\_type.findViewById(R.id.***text\_jie\_content***);  
 TextView text\_jie\_title = layout\_type.findViewById(R.id.***text\_jie\_title***);  
 **if** (type == 1) {  
 text\_jie\_title.setText(**"冻结设备"**);  
 text\_jie\_content.setText(**"是否确认冻结设备？"**);  
 btn\_jie\_sure.setText(**"确认冻结"**);  
 } **else if** (type == 2) {  
 text\_jie\_title.setText(**"解除绑定人员"**);  
 text\_jie\_content.setText(**"是否确认解除绑定人员？"**);  
 btn\_jie\_sure.setText(**"确认解绑"**);  
 } **else if** (type == 3) {  
 text\_jie\_title.setText(**"解冻设备"**);  
 text\_jie\_content.setText(**"是否确认解冻设备？"**);  
 btn\_jie\_sure.setText(**"确认解冻"**);  
 }  
 btn\_jie\_sure.setOnClickListener(v -> {  
 **if** (type == 1 || type == 3) {  
 **try** {  
 **blockDialog**.show();  
 postChange();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 } **else if** (type == 2) {  
 **try** {  
 **blockDialog**.show();  
 postJie();  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 **dialog\_tips**.dismiss();  
 });  
 btn\_jie\_cancel.setOnClickListener(v -> {  
 **dialog\_tips**.dismiss();  
 });  
 }  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 *// 扫描二维码/条码回传* **if** (requestCode == **REQUEST\_CODE\_SCAN** && resultCode == ***RESULT\_OK***) {  
 **if** (data != **null**) {  
 String content = data.getStringExtra(Constant.***CODED\_CONTENT***);  
 Log.*i*(**"toby"**, **"handleMessage: "** + content);  
 **try** {  
 JSONObject object = **new** JSONObject(content);  
 String id = object.getString(**"id"**);  
 String uid = object.getString(**"uid"**);  
 **blockDialog**.show();  
 postPhone(id,uid);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
  
 **static class** bangViewHolder {  
 @BindView(R.id.***edit\_bang\_number***)  
 TextView **editBangNumber**;  
 @BindView(R.id.***text\_bang\_color***)  
 TextView **textBangColor**;  
 @BindView(R.id.***relative\_bang\_color***)  
 RelativeLayout **relativeBangColor**;  
 @BindView(R.id.***edit\_bang\_name***)  
 EditText **editBangName**;  
 @BindView(R.id.***edit\_bang\_phone***)  
 EditText **editBangPhone**;  
 @BindView(R.id.***text\_bang\_type***)  
 TextView **textBangType**;  
 @BindView(R.id.***relative\_bang\_type***)  
 RelativeLayout **relativeBangType**;  
 @BindView(R.id.***btn\_add\_sure***)  
 Button **btnAddSure**;  
  
 bangViewHolder(View view) {  
 ButterKnife.*bind*(**this**, view);  
 }  
 }  
  
 **static class** addViewHolder {  
 @BindView(R.id.***edit\_add\_number***)  
 EditText **editAddNumber**;  
 @BindView(R.id.***text\_add\_color***)  
 TextView **textAddColor**;  
 @BindView(R.id.***relative\_add\_color***)  
 RelativeLayout **relativeAddColor**;  
 @BindView(R.id.***edit\_add\_name***)  
 EditText **editAddName**;  
 @BindView(R.id.***text\_add\_sex***)  
 TextView **textAddSex**;  
 @BindView(R.id.***relative\_add\_sex***)  
 RelativeLayout **relativeAddSex**;  
 @BindView(R.id.***edit\_add\_phone***)  
 EditText **editAddPhone**;  
 @BindView(R.id.***text\_add\_type***)  
 TextView **textAddType**;  
 @BindView(R.id.***relative\_add\_type***)  
 RelativeLayout **relativeAddType**;  
 @BindView(R.id.***btn\_add\_sure***)  
 Button **btnAddSure**;  
  
 addViewHolder(View view) {  
 ButterKnife.*bind*(**this**, view);  
 }  
 }  
}

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:clipToPadding="true"  
 android:fitsSystemWindows="true"  
 android:orientation="vertical"**>  
  
 <**include  
 android:id="@+id/rl\_title"  
 layout="@layout/layout\_main\_title"** />  
  
 <**include layout="@layout/layout\_main"**/>  
 </**LinearLayout**>  
  
 <**android.support.design.widget.NavigationView  
 android:id="@+id/nav"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="left"  
 android:fitsSystemWindows="true"**>  
  
 <**include layout="@layout/layout\_main\_left"** />  
 </**android.support.design.widget.NavigationView**>  
  
</**android.support.v4.widget.DrawerLayout**>

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/white"  
 android:orientation="vertical"**>  
  
 <**include layout="@layout/item\_top"** />  
  
 <**EditText  
 android:id="@+id/edit\_edit\_old"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/dp\_45"  
 android:layout\_marginTop="@dimen/dp\_35"  
 android:background="@color/transparent"  
 android:hint="请输入旧密码"  
 android:inputType="textPassword"  
 android:paddingLeft="@dimen/dp\_15"  
 android:paddingRight="@dimen/dp\_15"  
 android:textColor="@color/black"  
 android:textColorHint="@color/grey\_hint"  
 android:textSize="@dimen/sp\_14"** />  
  
 <**View  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/dp\_1"  
 android:layout\_marginLeft="@dimen/dp\_15"  
 android:layout\_marginRight="@dimen/dp\_15"  
 android:background="@color/all\_bg"** />  
  
 <**EditText  
 android:id="@+id/edit\_edit\_new"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/dp\_45"  
 android:layout\_marginTop="@dimen/dp\_25"  
 android:background="@color/transparent"  
 android:hint="请输入新密码"  
 android:inputType="textPassword"  
 android:paddingLeft="@dimen/dp\_15"  
 android:paddingRight="@dimen/dp\_15"  
 android:textColor="@color/black"  
 android:textColorHint="@color/grey\_hint"  
 android:textSize="@dimen/sp\_14"** />  
  
 <**View  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/dp\_1"  
 android:layout\_marginLeft="@dimen/dp\_15"  
 android:layout\_marginRight="@dimen/dp\_15"  
 android:background="@color/all\_bg"** />  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="@dimen/dp\_25"  
 android:paddingLeft="@dimen/dp\_15"  
 android:paddingRight="@dimen/dp\_15"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginRight="@dimen/dp\_15"  
 android:layout\_toLeftOf="@+id/text\_edit\_code"  
 android:orientation="vertical"**>  
  
 <**EditText  
 android:id="@+id/edit\_edit\_code"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/dp\_45"  
 android:background="@color/transparent"  
 android:hint="请输入验证码"  
 android:textColor="@color/black"  
 android:textColorHint="@color/grey\_hint"  
 android:textSize="@dimen/sp\_14"** />  
  
 <**View  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/dp\_1"  
 android:background="@color/all\_bg"** />  
  
 </**LinearLayout**>  
  
 <**TextView  
 android:id="@+id/text\_edit\_code"  
 android:layout\_width="@dimen/dp\_80"  
 android:layout\_height="@dimen/dp\_30"  
 android:layout\_alignParentRight="true"  
 android:layout\_centerVertical="true"  
 android:background="@drawable/btn\_green"  
 android:gravity="center"  
 android:text="获取验证码"  
 android:textColor="@color/white"  
 android:textSize="@dimen/sp\_12"** />  
  
 </**RelativeLayout**>  
  
 <**TextView  
 android:id="@+id/text\_edit\_count"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true"  
 android:layout\_centerVertical="true"  
 android:layout\_margin="@dimen/dp\_15"  
 android:lineSpacingExtra="@dimen/dp\_5"  
 android:textColor="@color/grey\_hint"  
 android:textSize="@dimen/sp\_12"** />  
  
 <**Button  
 android:id="@+id/btn\_edit\_sure"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/dp\_40"  
 android:layout\_marginLeft="@dimen/dp\_35"  
 android:layout\_marginRight="@dimen/dp\_35"  
 android:layout\_marginTop="@dimen/dp\_20"  
 android:background="@drawable/btn\_green"  
 android:text="确认修改"  
 android:textColor="@color/white"  
 android:textSize="@dimen/sp\_15"** />  
  
</**LinearLayout**>

**public class** AddAddressFragment **extends** BaseFragment {  
  
 @BindView(R.id.***text\_package\_been***)  
 TextView **textPackageBeen**;  
 @BindView(R.id.***text\_package\_haveto***)  
 TextView **textPackageHaveto**;  
 @BindView(R.id.***viewpager***)  
 NoScrollViewPager **viewpager**;  
 Unbinder **unbinder**;  
 **public static** String *id* = **""**;  
 **public static int** *num*;  
 @BindView(R.id.***view\_package\_been***)  
 View **viewPackageBeen**;  
 @BindView(R.id.***view\_package\_haveto***)  
 View **viewPackageHaveto**;  
  
 @Override  
 **protected** View getSuccessView() {  
 View view = View.*inflate*(getActivity(), R.layout.***fragment\_addcity***, **null**);  
 ButterKnife.*bind*(**this**, view);  
 **if** (getArguments() != **null**) {  
 *num* = getArguments().getInt(**"num"**);  
 *id* = getArguments().getString(**"id"**, **""**);  
 }  
 setListener();  
 **isPrepared** = **true**;  
 InitViewPager();  
 **return** view;  
 }  
  
 @Override  
 **protected void** loadData() {  
 **if** (!**isPrepared** || !**isVisable**) {  
 **return**;  
 }  
 }  
  
 **public static** AddAddressFragment newInstance(String id, **int** num) {  
 Bundle args = **new** Bundle();  
 args.putString(**"id"**, id);  
 args.putInt(**"num"**, num);  
 AddAddressFragment fragment = **new** AddAddressFragment();  
 fragment.setArguments(args);  
 **return** fragment;  
 }  
  
 **private void** setListener() {  
 **textPackageBeen**.setOnClickListener(**this**);  
 **textPackageHaveto**.setOnClickListener(**this**);  
 }  
  
 **private void** InitViewPager() {  
 MainPagerAdapter pagerAdapter = **new** MainPagerAdapter(  
 getChildFragmentManager(), **""**) {  
  
 *//获取第position位置的Fragment* @Override  
 **public** Fragment getItem(**int** position) {  
 Fragment fragment = **null**;  
 **switch** (position) {  
 **case** 0:  
 fragment = AddDirectmailFragment.*newInstance*(1, *id*);  
 **break**;  
 **case** 1:  
 fragment = AddsFragment.*newInstance*(2, *id*);  
 **break**;  
 }  
 **return** fragment;  
 }  
  
 *//该方法的返回值i表明该Adapter总共包括多少个Fragment* @Override  
 **public int** getCount() {  
 **return** 2;  
 }  
 };  
  
 *//为ViewPager组件设置FragmentPagerAdapter* **viewpager**.setAdapter(pagerAdapter);  
 **if** (*num* == 2) {  
 **viewpager**.setCurrentItem(1);  
 **viewPackageBeen**.setVisibility(View.***INVISIBLE***);  
 **viewPackageHaveto**.setVisibility(View.***VISIBLE***);  
 **textPackageHaveto**.setTextColor(getResources().getColor(R.color.***main\_red***));  
 **textPackageBeen**.setTextColor(getResources().getColor(R.color.***main\_black***));  
 } **else** {  
 **viewpager**.setCurrentItem(0);  
 **viewPackageBeen**.setVisibility(View.***VISIBLE***);  
 **viewPackageHaveto**.setVisibility(View.***INVISIBLE***);  
 **textPackageHaveto**.setTextColor(getResources().getColor(R.color.***main\_black***));  
 **textPackageBeen**.setTextColor(getResources().getColor(R.color.***main\_red***));  
 }  
 }  
  
 @Override  
 **protected void** setActionBar() {  
 ClickButtonActivity activity = (ClickButtonActivity) getActivity();  
 activity.**textTop**.setText(R.string.***add\_add\_address***);  
 }  
  
 @Override  
 **public boolean** onBackPressed() {  
 **return false**;  
 }  
  
 @Override  
 **protected** Object requestData() {  
 **return** ConstantUtils.***STATE\_SUCCESSED***;  
 }  
  
 @Override  
 **public void** onDestroyView() {  
 **super**.onDestroyView();  
*// ButterKnife.unbind(this);* **unbinder**.unbind();  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***text\_package\_haveto***:  
 **viewpager**.setCurrentItem(1);  
 **viewPackageBeen**.setVisibility(View.***INVISIBLE***);  
 **viewPackageHaveto**.setVisibility(View.***VISIBLE***);  
 **textPackageHaveto**.setTextColor(getResources().getColor(R.color.***main\_red***));  
 **textPackageBeen**.setTextColor(getResources().getColor(R.color.***main\_black***));  
 **break**;  
 **case** R.id.***text\_package\_been***:  
 **viewpager**.setCurrentItem(0);  
 **viewPackageBeen**.setVisibility(View.***VISIBLE***);  
 **viewPackageHaveto**.setVisibility(View.***INVISIBLE***);  
 **textPackageHaveto**.setTextColor(getResources().getColor(R.color.***main\_black***));  
 **textPackageBeen**.setTextColor(getResources().getColor(R.color.***main\_red***));  
 **break**;  
  
 }  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 *//* ***TODO: inflate a fragment view*** View rootView = **super**.onCreateView(inflater, container, savedInstanceState);  
 **unbinder** = ButterKnife.*bind*(**this**, rootView);  
 **return** rootView;  
 }  
}

**public class** BalanceFragment **extends** BaseFragment {  
 Unbinder **unbinder**;  
 @BindView(R.id.linear\_balance\_back)  
 LinearLayout **linearBalanceBack**;  
 @BindView(R.id.text\_balance\_detail)  
 TextView **textBalanceDetail**;  
 @BindView(R.id.btn\_balance\_recharge)  
 Button **btnBalanceRecharge**;  
 @BindView(R.id.btn\_balance\_withdrawals)  
 Button **btnBalanceWithdrawals**;  
 @BindView(R.id.text\_balance\_rnb)  
 TextView **textBalanceRnb**;  
 @BindView(R.id.text\_balance\_ay)  
 TextView **textBalanceAy**;  
 @BindView(R.id.text\_balance\_choose)  
 ImageView **textBalanceChoose**;  
 **private** BlockDialog **blockDialog**;  
 **private** String **is\_auto\_moeny** = **""**;  
  
 @Override  
 **protected** View getSuccessView() {  
 View view = View.*inflate*(getActivity(), R.layout.fragment\_balance, **null**);  
 ButterKnife.*bind*(**this**, view);  
 **blockDialog** = **new** BlockDialog(getActivity());  
 setListener();  
 **isPrepared** = **true**;  
 **return** view;  
 }  
  
 @Override  
 **protected void** loadData() {  
 **if** (!**isPrepared** || !**isVisable**) {  
 **return**;  
 }  
 *//填充各控件的数据* }  
  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 **blockDialog**.show();  
 message();  
 }  
  
 **private void** setListener() {  
 **linearBalanceBack**.setOnClickListener(view -> getActivity().finish());  
 **textBalanceDetail**.setOnClickListener(**this**);  
 **btnBalanceRecharge**.setOnClickListener(**this**);  
 **btnBalanceWithdrawals**.setOnClickListener(view -> ToastUtil.*show*(**"功能暂未开发"**));  
 **textBalanceChoose**.setOnClickListener(view -> {  
 **if** (**is\_auto\_moeny**.equals(**"1"**)){  
 **is\_auto\_moeny** = **"0"**;  
 **blockDialog**.show();  
 message\_post();  
 }**else** {  
 **is\_auto\_moeny** = **"1"**;  
 **blockDialog**.show();  
 message\_post();  
 }  
 });  
 }  
  
 **private void** message() {  
 RequestCall call = SYApplication.*genericClient*()  
 .url(SYApplication.*path\_url* + **"/user/index/index"**).build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **blockDialog**.dismiss();  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 **boolean** status = jsonObject.getBoolean(**"status"**);  
 String msg = jsonObject.getString(**"msg"**);  
 **if** (status) {  
 JSONObject jsonData = jsonObject.getJSONObject(**"data"**);  
 String moeny\_aud = jsonData.getString(**"moeny\_aud"**);  
 **is\_auto\_moeny** = jsonData.getString(**"is\_auto\_moeny"**);  
 String moeny = jsonData.getString(**"moeny"**);  
 **textBalanceRnb**.setText(**"￥"**+moeny);  
 **textBalanceAy**.setText(**"A$"**+moeny\_aud);  
 **if** (**is\_auto\_moeny**.equals(**"1"**)){  
 **textBalanceChoose**.setImageResource(R.drawable.register\_gou);  
 }**else** {  
 **textBalanceChoose**.setImageResource(R.drawable.register\_not);  
 }  
  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 **private void** message\_post() {  
 RequestCall call = SYApplication.*postFormBuilder*()  
 .url(SYApplication.*path\_url* + **"/user/index/autoMoenyUpdate"**)  
 .addParams(**"is\_auto\_moeny"**,**is\_auto\_moeny**).build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **blockDialog**.dismiss();  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 **boolean** status = jsonObject.getBoolean(**"status"**);  
 String msg = jsonObject.getString(**"msg"**);  
 **if** (status) {  
 **if** (**is\_auto\_moeny**.equals(**"1"**)){  
 **textBalanceChoose**.setImageResource(R.drawable.register\_gou);  
 }**else** {  
 **textBalanceChoose**.setImageResource(R.drawable.register\_not);  
 }  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
 @Override  
 **protected void** setActionBar() {  
 ClickButtonActivity activity = (ClickButtonActivity) getActivity();  
 activity.**relativeTop**.setVisibility(View.***GONE***);  
 }  
  
 @Override  
 **public boolean** onBackPressed() {  
 **return false**;  
 }  
  
 @Override  
 **protected** Object requestData() {  
 **return** ConstantUtils.***STATE\_SUCCESSED***;  
 }  
  
 @Override  
 **public void** onDestroyView() {  
 **super**.onDestroyView();  
*// ButterKnife.unbind(this);* **unbinder**.unbind();  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 StartUtils.*startActivityById*(getActivity(), v.getId());  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 *//* ***TODO: inflate a fragment view*** View rootView = **super**.onCreateView(inflater, container, savedInstanceState);  
 **unbinder** = ButterKnife.*bind*(**this**, rootView);  
 **return** rootView;  
 }  
}

**public class** SettingFragment **extends** TakePhotoFragment **implements** View.OnClickListener {  
 Unbinder **unbinder**;  
 @BindView(R.id.***linear\_setting\_back***)  
 LinearLayout **linearSettingBack**;  
 @BindView(R.id.***image\_setting\_head***)  
 ImageView **imageSettingHead**;  
 @BindView(R.id.***edit\_setting\_nickname***)  
 EditText **editSettingNickname**;  
 @BindView(R.id.***text\_setting\_sex***)  
 TextView **textSettingSex**;  
 @BindView(R.id.***text\_setting\_add***)  
 TextView **textSettingAdd**;  
 @BindView(R.id.***relative\_setting\_sex***)  
 RelativeLayout **relativeSettingSex**;  
 @BindView(R.id.***relative\_setting\_head***)  
 RelativeLayout **relativeSettingHead**;  
 @BindView(R.id.***image\_setting\_zheng***)  
 ImageView **imageSettingZheng**;  
 @BindView(R.id.***image\_setting\_wen***)  
 ImageView **imageSettingWen**;  
 @BindView(R.id.***tui\_login***)  
 TextView **tuiLogin**;  
 **private** BlockDialog **blockDialog**;  
 **private** String **sex** = **""**, **nickname** = **""**, **avatar** = **""**, **sex\_copy** = **""**;  
 **private** ArrayList<TImage> **images**;  
 **private** InputMethodManager **inputManager**;  
 PopupWindow **popupWindow** = **null**;  
 **private** String **normal\_push** = **"1"**, **abnormal\_push** = **"1"**;  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 *//* ***TODO: inflate a fragment view*** View view = View.*inflate*(getActivity(), R.layout.***ftagment\_setting***, **null**);  
 **unbinder** = ButterKnife.*bind*(**this**, view);  
 **return** view;  
 }  
  
 @Override  
 **public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 setListener();  
 **blockDialog** = **new** BlockDialog(getActivity());  
 **blockDialog**.show();  
 message();  
 }  
  
 **private void** setListener() {  
 ClickButtonActivity activity = (ClickButtonActivity) getActivity();  
 activity.**relativeTop**.setVisibility(View.***GONE***);  
 **linearSettingBack**.setOnClickListener(**this**);  
 **textSettingAdd**.setOnClickListener(**this**);  
 **relativeSettingHead**.setOnClickListener(**this**);  
 **relativeSettingSex**.setOnClickListener(**this**);  
 **images** = **new** ArrayList<>();  
 **inputManager** = (InputMethodManager) getActivity().getSystemService(Context.***INPUT\_METHOD\_SERVICE***);  
 **imageSettingWen**.setOnClickListener(view -> {  
 **if** (**abnormal\_push**.equals(**"1"**)) {  
 **abnormal\_push** = **"0"**;  
 **imageSettingWen**.setImageResource(R.drawable.***pay\_close***);  
 } **else** {  
 **abnormal\_push** = **"1"**;  
 **imageSettingWen**.setImageResource(R.drawable.***pay\_open***);  
 }  
 });  
 **imageSettingZheng**.setOnClickListener(view -> {  
 **if** (**normal\_push**.equals(**"1"**)) {  
 **normal\_push** = **"0"**;  
 **imageSettingZheng**.setImageResource(R.drawable.***pay\_close***);  
 } **else** {  
 **normal\_push** = **"1"**;  
 **imageSettingZheng**.setImageResource(R.drawable.***pay\_open***);  
 }  
 });  
 }  
  
 **private void** message() {  
 RequestCall call = SYApplication.*genericClient*()  
 .url(SYApplication.*path\_url* + **"/user/index/edit"**).build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **blockDialog**.dismiss();  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 **boolean** status = jsonObject.getBoolean(**"status"**);  
 String msg = jsonObject.getString(**"msg"**);  
 **if** (status) {  
 JSONObject jsonData = jsonObject.getJSONObject(**"data"**);  
 String avatar = jsonData.getString(**"avatar"**);  
 String nickname = jsonData.getString(**"nickname"**);  
 String sex\_copy = jsonData.getString(**"sex\_copy"**);  
 **sex** = jsonData.getString(**"sex"**);  
 **abnormal\_push** = jsonData.getString(**"abnormal\_push"**);  
 **normal\_push** = jsonData.getString(**"normal\_push"**);  
 **editSettingNickname**.setText(nickname);  
 **textSettingSex**.setText(sex\_copy);  
 **if** (**abnormal\_push**.equals(**"0"**)) {  
 **imageSettingWen**.setImageResource(R.drawable.***pay\_close***);  
 } **else** {  
 **imageSettingWen**.setImageResource(R.drawable.***pay\_open***);  
 }  
 **if** (**normal\_push**.equals(**"0"**)) {  
 **imageSettingZheng**.setImageResource(R.drawable.***pay\_close***);  
 } **else** {  
 **imageSettingZheng**.setImageResource(R.drawable.***pay\_open***);  
 }  
 Glide\_Image.*load*(getActivity(), avatar, **imageSettingHead**, **new** CropCircleTransformation(getActivity()));  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 **private void** message\_post() {  
 PostFormBuilder builder = SYApplication.*postFormBuilder*()  
 .url(SYApplication.*path\_url* + **"/user/index/edit"**)  
 .addParams(**"sex"**, **sex**)  
 .addParams(**"nickname"**, **nickname**)  
 .addParams(**"normal\_push"**, **normal\_push**)  
 .addParams(**"abnormal\_push"**, **abnormal\_push**);  
  
 **if** (!**avatar**.equals(**""**)) {  
 builder.addFile(**"avatar"**, **avatar**, **new** File(**avatar**));  
 }  
 RequestCall call = builder.build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **blockDialog**.dismiss();  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 **boolean** status = jsonObject.getBoolean(**"status"**);  
 String msg = jsonObject.getString(**"msg"**);  
 **if** (status) {  
 ToastUtil.*show*(msg);  
 getActivity().finish();  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 @Override  
 **public void** onDestroyView() {  
 **super**.onDestroyView();  
*// ButterKnife.unbind(this);* **unbinder**.unbind();  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***linear\_setting\_back***:  
 getActivity().finish();  
 **break**;  
 **case** R.id.***text\_setting\_add***:  
 **nickname** = **editSettingNickname**.getText().toString();  
 **sex\_copy** = **textSettingSex**.getText().toString();  
 **blockDialog**.show();  
 message\_post();  
 **break**;  
 **case** R.id.***relative\_setting\_sex***:  
 hideKeyboard();  
 **blockDialog**.show();  
 message\_type();  
 **break**;  
 **case** R.id.***relative\_setting\_head***:  
 hideKeyboard();  
 **new** PopupWindow\_banner(getActivity(), **relativeSettingSex**);  
 **break**;  
 }  
 }  
  
 **private void** message\_type() {  
 RequestCall call = SYApplication.*genericClient*()  
 .url(SYApplication.*path\_url* + **"/common/category/get\_list"**)  
 .addParams(**"id"**, **"575"**).build();  
  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **blockDialog**.dismiss();  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 **boolean** status = jsonObject.getBoolean(**"status"**);  
 String msg = jsonObject.getString(**"msg"**);  
 **if** (status) {  
 String data = jsonObject.getString(**"data"**);  
 popwindow(data);  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 **private void** popwindow(String restles) {  
 View popview = LayoutInflater.*from*(getActivity()).inflate(  
 R.layout.***dialog\_list***, **null**);  
 **popupWindow** = **new** PopupWindow(popview, DrawerLayout.LayoutParams.***MATCH\_PARENT***,  
 DrawerLayout.LayoutParams.***WRAP\_CONTENT***);  
 **popupWindow**.setFocusable(**true**);  
 **popupWindow**.setOutsideTouchable(**true**);  
 setBackgroundAlpha(0.5f);  
 **popupWindow**.setBackgroundDrawable(**new** BitmapDrawable());  
 RecyclerView recycle\_city = popview  
 .findViewById(R.id.***recycle\_city***);  
 **final** ArrayList<HomeModel> modelArrayList;  
 PopBuyAdapter adapter;  
 LinearLayoutManager linearLayoutManager;  
 modelArrayList = **new** ArrayList<>();  
 adapter = **new** PopBuyAdapter(getActivity(), modelArrayList);  
 *//设置固定大小* recycle\_city.setHasFixedSize(**true**);  
 *//创建线性布局* linearLayoutManager = **new** LinearLayoutManager(getActivity());  
 *//垂直方向* linearLayoutManager.setOrientation(OrientationHelper.***VERTICAL***);  
 *//给RecyclerView设置布局管理器* recycle\_city.setLayoutManager(linearLayoutManager);  
 recycle\_city.setAdapter(adapter);  
 **try** {  
 JSONArray jsonData = **new** JSONArray(restles);  
 **for** (**int** i = 0; i < jsonData.length(); i++) {  
 HomeModel homeModel = **new** HomeModel();  
 String id = jsonData.getJSONObject(i).getString(**"id"**);  
 String value = jsonData.getJSONObject(i).getString(**"value"**);  
 homeModel.setId(id);  
 homeModel.setName(value);  
 modelArrayList.add(homeModel);  
 adapter.notifyDataSetChanged();  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block****// e.printStackTrace();* }  
 adapter.setOnItemClickListener((view, position) -> {  
 **popupWindow**.dismiss();  
 **sex** = modelArrayList.get(position).getId();  
 **textSettingSex**.setText(modelArrayList.get(position).getName());  
 });  
 **popupWindow**.setOnDismissListener(**new** poponDismissListener());  
 **if** (**popupWindow**.isShowing()) {  
 setBackgroundAlpha(0f);  
 **popupWindow**.dismiss();  
 } **else** {  
 **popupWindow**.showAtLocation(**textSettingSex**, Gravity.***BOTTOM***, 0, 0);  
 }  
 }  
  
 */\*\*  
 \* 添加新笔记时弹出的popWin关闭的事件，主要是为了将背景透明度改回来  
 \*/* **class** poponDismissListener **implements** PopupWindow.OnDismissListener {  
  
 @Override  
 **public void** onDismiss() {  
 *//* ***TODO Auto-generated method stub*** setBackgroundAlpha(1f);  
 }  
  
 }  
  
 */\*\*  
 \* 设置pop背景透明度  
 \*/* **public void** setBackgroundAlpha(**float** bgAlpha) {  
 WindowManager.LayoutParams lp = ((getActivity())).getWindow()  
 .getAttributes();  
 lp.**alpha** = bgAlpha;  
 getActivity().getWindow().setAttributes(lp);  
 }  
  
 **public void** hideKeyboard() {  
 **if** (**this** == **null**) **return**;  
 **if** (getActivity().getWindow().getAttributes().**softInputMode** != WindowManager.LayoutParams.***SOFT\_INPUT\_STATE\_HIDDEN***) {  
 **if** (getActivity().getCurrentFocus() != **null**)  
 **inputManager**.hideSoftInputFromWindow(getActivity().getCurrentFocus().getWindowToken(),  
 InputMethodManager.***HIDE\_NOT\_ALWAYS***);  
 }  
 }  
  
 **public class** PopupWindow\_banner **extends** PopupWindow {  
  
 **public** PopupWindow\_banner(Context mContext, View parent) {  
 View view = View  
 .*inflate*(mContext, R.layout.***item\_popupwindows***, **null**);  
 setWidth(ViewGroup.LayoutParams.***MATCH\_PARENT***);  
 setHeight(ViewGroup.LayoutParams.***MATCH\_PARENT***);  
 setBackgroundDrawable(**new** BitmapDrawable());  
 setFocusable(**true**);  
 setOutsideTouchable(**true**);  
 setContentView(view);  
 showAtLocation(parent, Gravity.***BOTTOM***, 0, 0);  
 update();  
  
 Button bt1 = view  
 .findViewById(R.id.***item\_popupwindows\_camera***);  
 Button bt2 = view  
 .findViewById(R.id.***item\_popupwindows\_Photo***);  
 Button bt3 = view  
 .findViewById(R.id.***item\_popupwindows\_cancel***);  
 **final** TakePhoto takePhoto = getTakePhoto();  
 File file = **new** File(Environment.*getExternalStorageDirectory*(), **"/temp/"** + System.*currentTimeMillis*() + **".jpg"**);  
 **if** (!file.getParentFile().exists()) file.getParentFile().mkdirs();  
 **final** Uri imageUri;  
 **if** (Build.VERSION.***SDK\_INT*** <= Build.VERSION\_CODES.***M***) {  
 imageUri = Uri.*fromFile*(file);  
 } **else** {  
 */\*\*  
 \* 7.0 调用系统相机拍照不再允许使用Uri方式，应该替换为FileProvider  
 \* 并且这样可以解决MIUI系统上拍照返回size为0的情况  
 \*/* imageUri = FileProvider.*getUriForFile*(getActivity(), ProviderUtil.*getFileProviderName*(getActivity()), file);  
 }  
 bt1.setOnClickListener(v -> {  
 takePhoto.onPickFromCaptureWithCrop(imageUri, getCropOptions());  
 dismiss();  
 });  
 bt2.setOnClickListener(v -> {  
 dismiss();  
 **int** limit = 1;  
 **if** (limit >= 1) {  
 takePhoto.onPickFromGalleryWithCrop(imageUri, getCropOptions());  
 }  
 });  
 bt3.setOnClickListener(v -> dismiss());  
  
 }  
 }  
  
 **private** CropOptions getCropOptions() {  
 **int** height = 512;  
 **int** width = 512;  
 **boolean** withWonCrop = **true**;  
 CropOptions.Builder builder = **new** CropOptions.Builder();  
 builder.setAspectX(width).setAspectY(height);  
 builder.setWithOwnCrop(withWonCrop);  
 **return** builder.create();  
 }  
  
 @Override  
 **public void** takeSuccess(TResult result) {  
 **super**.takeSuccess(result);  
 **images** = result.getImages();  
 compressPhotoSend(**images**);  
 }  
  
 **private void** compressPhotoSend(ArrayList<TImage> arrayList2) {  
 String newPath = **""**;  
 **for** (**int** i = 0; i < arrayList2.size(); i++) {  
 newPath = BitmapUtil.*compressImageAndSave*(  
 getActivity().getApplicationContext(), arrayList2.get(i).getOriginalPath(), **false**);  
 }  
 **avatar** = newPath;  
 Glide\_Image.*load*(getActivity(), **avatar**, **imageSettingHead**, **new** CropCircleTransformation(getActivity()));  
 }  
  
 @OnClick(R.id.***tui\_login***)  
 **public void** onViewClicked(View view) {  
 **switch** (view.getId()) {  
 **case** R.id.***tui\_login***:  
 show();  
 **break**;  
 }  
 }  
  
 **private void** show() {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(getActivity());  
 builder.setMessage(getResources().getString(R.string.***exit\_login\_ti***)).setTitle(getResources().getString(R.string.***prompt***))  
 .setPositiveButton(getResources().getString(R.string.***setting\_logout***), (dialog, which) -> {  
 SharedPreferences.Editor editor = getActivity().getSharedPreferences(**"userLogin"**, Context.***MODE\_PRIVATE***).edit();  
 editor.putString(**"token"**, **""**);  
 editor.apply();  
 startActivity(**new** Intent(getActivity(), LoginActivity.**class**));  
 getActivity().finish();  
 }).setNegativeButton(getResources().getString(R.string.***login\_cancel***), (dialog, which) -> {  
  
 }).show();  
 }  
}

**public class** EditorFragment **extends** BaseFragment {  
 Unbinder **unbinder**;  
 @BindView(R.id.***recycler\_editor***)  
 RecyclerView **recyclerEditor**;  
 @BindView(R.id.***text\_editor\_add***)  
 TextView **textEditorAdd**;  
 **private** Dialog **dialog\_tips**,**dialog\_success**;  
  
 @Override  
 **protected** View getSuccessView() {  
 View view = View.*inflate*(getActivity(), R.layout.***fragment\_editor***, **null**);  
 ButterKnife.*bind*(**this**, view);  
 setListener();  
 **isPrepared** = **true**;  
 **return** view;  
 }  
  
 @Override  
 **protected void** loadData() {  
 **if** (!**isPrepared** || !**isVisable**) {  
 **return**;  
 }  
 *//填充各控件的数据* }  
  
 @Override  
 **protected void** setActionBar() {  
 ClickButtonActivity activity = (ClickButtonActivity) getActivity();  
 activity.**textTop**.setText(R.string.***goods\_editor***);  
 }  
  
 @Override  
 **public boolean** onBackPressed() {  
 **return false**;  
 }  
  
  
 **private void** setListener() {  
 **textEditorAdd**.setOnClickListener(view -> dialog\_tip());  
 }  
  
 **private void** dialog\_tip() {  
 LayoutInflater inflater\_type = getActivity().getLayoutInflater();  
 View layout\_type = inflater\_type.inflate(R.layout.***dialog\_add\_goods***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(getActivity());  
 **dialog\_tips** = record\_type.create();  
 **dialog\_tips**.show();  
 **dialog\_tips**.setContentView(layout\_type);  
 **dialog\_tips**.setCanceledOnTouchOutside(**false**);*//* Button btn\_add\_sure = layout\_type  
 .findViewById(R.id.***btn\_add\_sure***);*//* ImageView image\_dialog\_close = layout\_type  
 .findViewById(R.id.***image\_dialog\_close***);*//* **dialog\_tips**.getWindow().clearFlags(WindowManager.LayoutParams.***FLAG\_ALT\_FOCUSABLE\_IM***);  
 image\_dialog\_close.setOnClickListener(v -> {  
 *//* ***TODO Auto-generated method stub* dialog\_tips**.dismiss();  
 });  
  
 btn\_add\_sure.setOnClickListener(v -> {  
 *//* ***TODO Auto-generated method stub*** dialog\_success();  
 **dialog\_tips**.dismiss();  
 });  
 }  
  
 **private void** dialog\_success() {  
 LayoutInflater inflater\_type = getActivity().getLayoutInflater();  
 View layout\_type = inflater\_type.inflate(R.layout.***dialog\_add\_success***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(getActivity());  
 **dialog\_success** = record\_type.create();  
 **dialog\_success**.getWindow().setBackgroundDrawableResource(android.R.color.***transparent***);*//设置Dialog背景透明效果* **dialog\_success**.show();  
 **dialog\_success**.setContentView(layout\_type);  
 **dialog\_success**.setCanceledOnTouchOutside(**true**);*//* Button btn\_dialog\_add = layout\_type  
 .findViewById(R.id.***btn\_dialog\_add***);*//* Button btn\_dialog\_cancel = layout\_type  
 .findViewById(R.id.***btn\_dialog\_cancel***);  
  
 btn\_dialog\_cancel.setOnClickListener(v -> {  
 *//* ***TODO Auto-generated method stub* dialog\_success**.dismiss();  
 });  
  
 btn\_dialog\_add.setOnClickListener(v -> {  
 *//* ***TODO Auto-generated method stub* dialog\_success**.dismiss();  
 dialog\_tip();  
 });  
 }  
  
 @Override  
 **protected** Object requestData() {  
 **return** ConstantUtils.***STATE\_SUCCESSED***;  
 }  
  
 @Override  
 **public void** onDestroyView() {  
 **super**.onDestroyView();  
*// ButterKnife.unbind(this);* **unbinder**.unbind();  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 StartUtils.*startActivityById*(getActivity(), v.getId());  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 *//* ***TODO: inflate a fragment view*** View rootView = **super**.onCreateView(inflater, container, savedInstanceState);  
 **unbinder** = ButterKnife.*bind*(**this**, rootView);  
 **return** rootView;  
 }  
}

**public class** StorageFragment **extends** BaseFragment {  
  
 Unbinder **unbinder**;  
 @BindView(R.id.***recycler\_transfer***)  
 RecyclerView **recyclerTransfer**;  
 @BindView(R.id.***refresh\_btn***)  
 Button **refreshBtn**;  
 @BindView(R.id.***refresh\_linear***)  
 LinearLayout **refreshLinear**;  
 @BindView(R.id.***linear\_no\_data***)  
 LinearLayout **linearNoData**;  
 @BindView(R.id.***table\_layout***)  
 TabLayout **tabLayout**;  
  
 **private** ArrayList<HomeModel> **arrayList**;  
 **private** StorageAdapter **adapter**;  
 **private** String **type** = **""**, **stauts** = **""**;  
 **private** BlockDialog **blockDialog**;  
 **private** String **pageSize** = **"10"**;  
 **private int pageNo** = 1, **position**;  
 **private** String **order\_sn** = **""**, **status** = **""**;  
 **private** Dialog **dialog\_tips**, **dialog\_error**, **dialog\_wuliu**;  
 **private static** StorageFragment *localFragment*;  
  
 @Override  
 **protected** View getSuccessView() {  
 View view = View.*inflate*(getActivity(), R.layout.***fragment\_local\_second***, **null**);  
 ButterKnife.*bind*(**this**, view);  
 **if** (getArguments() != **null**) {  
 **type** = getArguments().getString(**"type"**);  
 **stauts** = getArguments().getString(**"states"**);  
 }  
 *localFragment* = **this**;  
 **blockDialog** = **new** BlockDialog(getActivity());  
 initView();  
 **isPrepared** = **true**;  
 **tabLayout**.setVisibility(View.***GONE***);  
 **return** view;  
 }  
  
 @Override  
 **protected void** loadData() {  
 **if** (!**isPrepared** || !**isVisable**) {  
 **return**;  
 }  
 *//填充各控件的数据* **arrayList**.clear();  
 **blockDialog**.show();  
 message();  
 }  
  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 loadData();  
 }  
  
 **public static** StorageFragment newInstance(String type, String states) {  
 Bundle args = **new** Bundle();  
 args.putString(**"type"**, type);  
 args.putString(**"states"**, states);  
 StorageFragment fragment = **new** StorageFragment();  
 fragment.setArguments(args);  
 **return** fragment;  
 }  
  
 @Override  
 **protected void** setActionBar() {  
 }  
  
 **private void** message() {  
 RequestCall call = SYApplication.*genericClient*()  
 .url(SYApplication.*path\_url* + **"/user/orders/lists"**)  
 .addParams(**"pageNo"**, **pageNo** + **""**)  
 .addParams(**"pageSize"**, **pageSize**)  
 .addParams(**"type"**, **type**)  
 .addParams(**"status"**, **stauts**)  
 .build().connTimeOut(100000).readTimeOut(100000);  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 **recyclerTransfer**.setVisibility(View.***GONE***);  
 **refreshLinear**.setVisibility(View.***VISIBLE***);  
 Log.*i*(**"toby"**, **"onError: "** + e);  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **blockDialog**.dismiss();  
 **recyclerTransfer**.setVisibility(View.***VISIBLE***);  
 **refreshLinear**.setVisibility(View.***GONE***);  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 String msg = jsonObject.getString(**"msg"**);  
 **int** code = jsonObject.getInt(**"code"**);  
 **if** (code == 501) {  
 SYApplication.*loginOut*();  
 Intent intent = **new** Intent(getActivity(), LoginActivity.**class**);  
 startActivity(intent);  
 getActivity().finish();  
 }  
 **if** (code == 200) {  
 JSONObject jsonData = jsonObject.getJSONObject(**"data"**);  
 JSONArray list = jsonData.getJSONArray(**"list"**);  
 **if** (list.length() == 0) {  
 **if** (**pageNo** == 1) {  
 **recyclerTransfer**.setVisibility(View.***GONE***);  
 **linearNoData**.setVisibility(View.***VISIBLE***);  
 }  
 } **else** {  
 **recyclerTransfer**.setVisibility(View.***VISIBLE***);  
 **linearNoData**.setVisibility(View.***GONE***);  
 }  
 **for** (**int** i = 0; i < list.length(); i++) {  
 HomeModel model = **new** HomeModel();  
 String order\_sn = list.getJSONObject(i).getString(**"order\_sn"**);  
 String logistics\_name = list.getJSONObject(i).getString(**"logistics\_name"**);  
 String status\_type = list.getJSONObject(i).getString(**"status\_type"**);  
 String title = list.getJSONObject(i).getString(**"storage\_transport\_sn"**);  
 String goodsList = list.getJSONObject(i).getString(**"goodsList"**);  
 String status\_time\_copy = list.getJSONObject(i).getString(**"status\_time\_copy"**);  
 String status\_copy = list.getJSONObject(i).getString(**"status\_copy"**);  
 String issue\_status = list.getJSONObject(i).getString(**"issue\_status"**);  
 model.setOrder\_sn(order\_sn);  
 model.setIssue\_status(issue\_status);  
 model.setName(logistics\_name);  
 model.setAddress(goodsList);  
 model.setTransport\_sn(title);  
 model.setTitle(status\_copy);  
 model.setStauts(status\_type);  
 model.setTime(status\_time\_copy);  
  
 **arrayList**.add(model);  
 **adapter**.notifyDataSetChanged();  
 }  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
  
  
 }  
 }  
 });  
 }  
  
 **private void** message\_wu() {  
 RequestCall call = SYApplication.*genericClient*()  
 .url(SYApplication.*path\_url* + **"/index/Logistics/index"**)  
 .addParams(**"type"**, **"1"**)  
 .addParams(**"order\_sn"**, **order\_sn**)  
 .addParams(**"is\_push"**, **"0"**).build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **blockDialog**.dismiss();  
 Log.*i*(**"toby"**, **"onResponse: "** + response);  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 String msg = jsonObject.getString(**"msg"**);  
 **boolean** status = jsonObject.getBoolean(**"status"**);  
 **if** (status) {  
 String jsonData = jsonObject.getString(**"data"**);  
 dialog\_wuliu(jsonData);  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 **private void** dialog\_wuliu(String data) {  
 LayoutInflater inflater\_type = getActivity().getLayoutInflater();  
 View layout\_type = inflater\_type.inflate(R.layout.***dialog\_wuliu***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(getActivity(),  
 R.style.***myNewsDialogStyle***);  
 **dialog\_wuliu** = record\_type.create();  
 **dialog\_wuliu**.show();  
 **dialog\_wuliu**.setContentView(layout\_type);  
 **dialog\_wuliu**.setCanceledOnTouchOutside(**true**);*//  
// 将对话框的大小按屏幕大小的百分比设置* WindowManager windowManager = getActivity().getWindowManager();  
 Display display = windowManager.getDefaultDisplay();  
 WindowManager.LayoutParams lp = **dialog\_wuliu**.getWindow().getAttributes();  
 lp.**width** = (**int**) (display.getWidth() \* 0.8); *//设置宽度* **dialog\_wuliu**.getWindow().setAttributes(lp);  
 TextView text\_wuliu\_number = layout\_type.findViewById(R.id.***text\_wuliu\_number***);  
 TextView text\_wuliu\_copy = layout\_type.findViewById(R.id.***text\_wuliu\_copy***);  
 RecyclerView recycler\_wuliu = layout\_type.findViewById(R.id.***recycler\_wuliu***);  
 ImageView image\_wuliu\_close = layout\_type.findViewById(R.id.***image\_wuliu\_close***);  
  
 **final** ArrayList<FlowTrace> modelArrayList;  
 JoinFlowAdapter adapter;  
 LinearLayoutManager linearLayoutManager;  
 modelArrayList = **new** ArrayList<>();  
 adapter = **new** JoinFlowAdapter(getActivity(), modelArrayList);  
 *//设置固定大小* recycler\_wuliu.setHasFixedSize(**true**);  
 *//创建线性布局* linearLayoutManager = **new** LinearLayoutManager(getActivity());  
 *//垂直方向* linearLayoutManager.setOrientation(OrientationHelper.***VERTICAL***);  
 *//给RecyclerView设置布局管理器* recycler\_wuliu.setLayoutManager(linearLayoutManager);  
 recycler\_wuliu.setAdapter(adapter);  
 **try** {  
 JSONObject jsonData = **new** JSONObject(data);  
 String number = jsonData.getString(**"number"**);  
 text\_wuliu\_number.setText(number);  
 JSONArray array = jsonData.getJSONArray(**"list"**);  
 **for** (**int** i = 0; i < array.length(); i++) {  
 FlowTrace trace = **new** FlowTrace();  
 trace.setNum(array.getJSONObject(i).getString(**"time"**));  
 trace.setAcceptStation(array.getJSONObject(i).getString(**"remark"**));  
 modelArrayList.add(trace);  
 adapter.notifyDataSetChanged();  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block****// e.printStackTrace();* }  
 image\_wuliu\_close.setOnClickListener(v -> {  
 *//* ***TODO Auto-generated method stub* dialog\_wuliu**.dismiss();  
 });  
 text\_wuliu\_copy.setOnClickListener(view -> {  
 copy(text\_wuliu\_number.getText().toString());  
 });  
  
 }  
  
  
 **private void** copy(String copy) {  
 ClipboardManager cm = (ClipboardManager) getActivity()  
 .getSystemService(Context.***CLIPBOARD\_SERVICE***);  
 ClipData myClip;  
 myClip = ClipData.*newPlainText*(**"text"**, copy);  
 cm.setPrimaryClip(myClip);  
 ToastUtil.*show*(**"复制成功，可以发给朋友们了。"**);  
 }  
  
 @Override  
 **public boolean** onBackPressed() {  
 **return false**;  
 }  
  
 **private void** initView() {  
 *//设置固定大小* **recyclerTransfer**.setHasFixedSize(**true**);  
 LinearLayoutManager mLayoutManager = **new** LinearLayoutManager(getActivity());  
 *//垂直方向* mLayoutManager.setOrientation(OrientationHelper.***VERTICAL***);  
 mLayoutManager.setSmoothScrollbarEnabled(**true**);  
 mLayoutManager.setAutoMeasureEnabled(**true**);  
  
 *//给RecyclerView设置布局管理器* **recyclerTransfer**.setLayoutManager(mLayoutManager);  
 **recyclerTransfer**.setHasFixedSize(**true**);  
 **recyclerTransfer**.setNestedScrollingEnabled(**false**);  
 **arrayList** = **new** ArrayList<>();  
 **adapter** = **new** StorageAdapter(getActivity(), **arrayList**, **clickListener**);  
 **recyclerTransfer**.setAdapter(**adapter**);  
  
 **adapter**.setOnItemClickListener((view, position) ->  
 StartUtils.*startActivityByIds*(getActivity(), -11, **arrayList**.get(position).getOrder\_sn(), **arrayList**.get(position).getStauts())  
 );  
 **refreshBtn**.setOnClickListener(view -> {  
 **blockDialog**.show();  
 message();  
 });  
  
 }  
  
 **public static void** refush() {  
 *localFragment*.**arrayList**.clear();  
 *localFragment*.**blockDialog**.show();  
 *localFragment*.**pageNo** = 1;  
 *localFragment*.message();  
 }  
  
 **public static void** loadMore() {  
 *localFragment*.**blockDialog**.show();  
 *localFragment*.**pageNo**++;  
 *localFragment*.message();  
 }  
  
 **private** View.OnClickListener **clickListener** = **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 **switch** (view.getId()) {  
 **case** R.id.***btn\_library\_return***:*//退件* **position** = (Integer) view.getTag();  
 **order\_sn** = **arrayList**.get(**position**).getOrder\_sn();  
 dialog\_tip(1);  
 **break**;  
 **case** R.id.***btn\_storage\_editor***:  
 **position** = (Integer) view.getTag();  
 StartUtils.*startActivityByIds*(getActivity(), view.getId(), **arrayList**.get(**position**).getOrder\_sn());  
  
 **break**;  
  
 **case** R.id.***btn\_library\_error***: *// 异常处理* **position** = (Integer) view.getTag();  
 HomeModel model=**arrayList**.get(**position**);  
 **order\_sn** = model.getOrder\_sn();  
 dialog\_error(**order\_sn**,model.getIssue\_status());  
  
 **break**;  
  
 **case** R.id.***btn\_library\_delete***:  
 **position** = (Integer) view.getTag();  
 **order\_sn** = **arrayList**.get(**position**).getOrder\_sn();  
 dialog\_tip(2);  
 **break**;  
 **case** R.id.***text\_storage\_copy***:*//物流查询* **position** = (Integer) view.getTag();  
 **order\_sn** = **arrayList**.get(**position**).getOrder\_sn();  
 **blockDialog**.show();  
 message\_wu();  
 **break**;  
 }  
  
 }  
 };  
  
 */\*\*  
 \* 国际转运异常商品处理  
 \** ***@param order*** *订单号  
 \** ***@param num*** *状态码  
 \*/* **private void** dialog\_error(String order,**final** String num) {  
 LayoutInflater inflater\_type = getActivity().getLayoutInflater();  
 View layout\_type = inflater\_type.inflate(R.layout.***dialog\_error***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(getActivity());  
 **dialog\_error** = record\_type.create();  
 **dialog\_error**.show();  
 **dialog\_error**.setContentView(layout\_type);  
 **dialog\_error**.setCanceledOnTouchOutside(**false**);  
 TextView text\_error\_sure = layout\_type.findViewById(R.id.***text\_error\_sure***);  
 TextView text\_error\_cancel = layout\_type.findViewById(R.id.***text\_error\_cancel***);  
 TextView text\_error\_content = layout\_type.findViewById(R.id.***text\_error\_content***);  
  
 **switch** (num){  
 **case "17"**:  
 String s1=getResources().getString(R.string.***error\_out***);  
 text\_error\_content.setText(order+**" "**+s1);  
 text\_error\_sure.setText(getResources().getString(R.string.***error\_again\_pack***));  
 text\_error\_cancel.setText(getResources().getString(R.string.***error\_not\_send***));  
 **break**;  
  
 **case "18"**:  
 String s2=getResources().getString(R.string.***tui\_jian***);  
 text\_error\_content.setText(order+**" "**+s2);  
 text\_error\_sure.setText(getResources().getString(R.string.***sure***));  
 text\_error\_cancel.setVisibility(View.***GONE***);  
 text\_error\_cancel.setText(R.string.***error\_not\_send***);  
 **break**;  
  
 **case "19"**:  
 String s3=getResources().getString(R.string.***hai\_jian***);  
 text\_error\_content.setText(order+**" "**+s3);  
 text\_error\_sure.setText(getResources().getString(R.string.***sure***));  
 text\_error\_cancel.setVisibility(View.***GONE***);  
 text\_error\_cancel.setText(R.string.***error\_not\_send***);  
  
 **break**;  
  
 **case "20"**:  
 String s4=getResources().getString(R.string.***pack\_back***);  
 text\_error\_content.setText(order+**" "**+s4);  
 text\_error\_sure.setText(getResources().getString(R.string.***sure***));  
 text\_error\_cancel.setVisibility(View.***GONE***);  
 text\_error\_cancel.setText(R.string.***error\_not\_send***);  
 **break**;  
  
 **case "21"**:  
 String s5=getResources().getString(R.string.***out\_time***);  
 text\_error\_content.setText(order+**" "**+s5);  
 text\_error\_sure.setText(getResources().getString(R.string.***sure***));  
 text\_error\_cancel.setVisibility(View.***GONE***);  
 text\_error\_cancel.setText(getResources().getString(R.string.***error\_not\_send***));  
 **break**;  
  
 **case "22"**:  
 String s6=getResources().getString(R.string.***other\_qian***);  
 text\_error\_content.setText(order+**" "**+s6);  
 text\_error\_sure.setText(R.string.***get\_finish***);  
 text\_error\_cancel.setText(R.string.***get\_no\_finish***);  
 **break**;  
  
 **case "23"**:  
 String s7=getResources().getString(R.string.***dou\_shi\_or\_posun***);  
 text\_error\_content.setText(order+**" "**+s7);  
 text\_error\_sure.setText(R.string.***question\_finish***);  
 text\_error\_cancel.setText(R.string.***error\_btn5***);  
  
 **break**;  
  
 **case "24"**:  
 String s8=getResources().getString(R.string.***tui\_finish***);  
 text\_error\_content.setText(order+**" "**+s8);  
 text\_error\_sure.setText(R.string.***sure***);  
 text\_error\_cancel.setVisibility(View.***GONE***);  
 text\_error\_cancel.setText(R.string.***error\_btn5***);  
 **break**;  
  
 **case "25"**:  
 String s9=getResources().getString(R.string.***my\_qian***);  
 text\_error\_content.setText(order+**" "**+s9);  
 text\_error\_sure.setText(getResources().getString(R.string.***sure***));  
 text\_error\_cancel.setVisibility(View.***GONE***);  
 text\_error\_cancel.setText(R.string.***login\_cancel***);  
 **break**;  
  
 **case "26"**:  
 String s10=getResources().getString(R.string.***update\_id\_card***);  
 text\_error\_content.setText(order+**" "**+s10);  
 text\_error\_sure.setText(R.string.***package\_add***);  
 text\_error\_cancel.setText(R.string.***login\_cancel***);  
  
 **break**;  
  
 **case "27"**:  
 String s11=getResources().getString(R.string.***wait\_pay\_money***);  
 text\_error\_content.setText(order+**" "**+s11);  
 text\_error\_sure.setText(R.string.***package\_add***);  
 text\_error\_cancel.setText(R.string.***login\_cancel***);  
 **break**;  
  
 **case "28"**:  
  
 String s12=getResources().getString(R.string.***other***);  
 text\_error\_content.setText(order+**" "**+s12);  
 text\_error\_sure.setText(R.string.***package\_add***);  
 text\_error\_cancel.setText(R.string.***login\_cancel***);  
  
 **break**;  
  
 }  
  
 text\_error\_sure.setOnClickListener(view -> {  
 **switch** (num){  
 **case "17"**:  
 */// 重新打包* againPackOrNotSend(2,order);  
 **dialog\_error**.dismiss();  
  
 **break**;  
 **case "22"**:  
 getOrLo(1,order);  
 **dialog\_error**.dismiss();  
  
 **break**;  
 **case "23"**:  
  
 upOrFinish(order,1,**null**);  
 **dialog\_error**.dismiss();  
 **break**;  
  
 **default**:  
 **dialog\_error**.dismiss();  
 **break**;  
 }  
  
 });  
  
 text\_error\_cancel.setOnClickListener(view -> {  
 **switch** (num){  
 **case "17"**:  
 *// 不发了，退件* againPackOrNotSend(1,order);  
 **dialog\_error**.dismiss();  
  
 **break**;  
  
 **case "22"**:  
 getOrLo(2,order);  
 **dialog\_error**.dismiss();  
  
 **break**;  
  
 **case "23"**:  
  
 startActivityForResult(**new** Intent(getActivity(),AddPicActivity.**class**),1234);  
  
 **dialog\_error**.dismiss();  
  
 **break**;  
  
 **default**:  
 **break**;  
 }  
 });  
  
*//  
// if (num.equals("17") || num.equals("18")) {  
// text\_error\_content.setText(R.string.error\_2);  
// text\_error\_sure.setText(R.string.error\_btn2);  
// text\_error\_cancel.setText(R.string.error\_btn3);  
// } else if (num.equals("22")) {  
// text\_error\_content.setText(R.string.error\_5);  
// text\_error\_sure.setText(R.string.error\_btn5);  
// text\_error\_cancel.setText(R.string.error\_btn6);  
// } else {  
// text\_error\_content.setText(R.string.error\_412);  
// text\_error\_sure.setText(R.string.error\_btn412);  
// text\_error\_cancel.setText(R.string.error\_btn413);  
// }  
// text\_error\_cancel.setOnClickListener(v -> {  
// dialog\_error.dismiss();  
// if (num.equals("17") || num.equals("18")) {  
// status = "18";  
// blockDialog.show();  
// message\_error();  
// } else if (num.equals("22")) {  
// status = "1";  
// blockDialog.show();  
// message\_error();  
// } else {  
// status = "23";  
// blockDialog.show();  
// message\_error();  
// }  
// });  
// text\_error\_sure.setOnClickListener(v -> {  
// //* ***TODO Auto-generated method stub****// dialog\_error.dismiss();  
// if (num.equals("17") || num.equals("18")) {  
// status = "1";  
// blockDialog.show();  
// message\_error();  
// } else if (num.equals("22")) {  
//// status = "1";  
// StartUtils.startActivityByIds(getActivity(), R.id.text\_error\_sure, order\_sn);  
// } else {  
// status = "1";  
// blockDialog.show();  
// message\_error();  
// }  
// });* }  
  
 @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if**(requestCode==1234&&resultCode== Activity.***RESULT\_OK***){  
 ArrayList<String> list=data.getStringArrayListExtra(**"list"**);  
 upOrFinish(**order\_sn**,2,list);  
  
 }  
 }  
  
 */\*\*  
 \* 完成 或者 上传开箱图片  
 \** ***@param order*** *\** ***@param code*** *问题已解决 2 上传开箱照片  
 \*/* **private void** upOrFinish(String order, **int** code, ArrayList<String> list) {  
 PostFormBuilder post=SYApplication.*postFormBuilder*();  
 **if**(code==2){  
 **if**(list!=**null**) {  
 **int** len = list.size();  
 **for** (**int** i = 0; i < len; i++) {  
 String pathStr = list.get(i);  
 File file=**new** File(pathStr);  
 post.addFile(**"box\_ablum"**, file.getName(), file);  
 }  
 }  
 }  
  
 RequestCall call = post.url(SYApplication.*path\_url* + **"/user/orders/problemPackage23"**)  
 .addParams(**"order\_sn"**, order)  
 .addParams(**"type"**, String.*valueOf*(code)).build().connTimeOut(100000).readTimeOut(100000);  
  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 Log.*i*(**"result"**,**"上传开箱图片------"**+response);  
 **try** {  
 JSONObject json = **new** JSONObject(response);  
 String msg = json.getString(**"msg"**);  
 **int** code = json.getInt(**"code"**);  
 **if** (code == 200) {  
 ToastUtil.*show*(msg);  
  
 **arrayList**.clear();  
 */// 重新获取数据* message();  
  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
 }  
  
 */\*\*  
 \* 收到或者丢失  
 \** ***@param i*** *\** ***@param order*** *\*/* **private void** getOrLo(**int** i, String order) {  
 RequestCall call = SYApplication.*postFormBuilder*()  
 .url(SYApplication.*path\_url* + **"/user/orders/problemPackage22"**)  
 .addParams(**"order\_sn"**, order)  
 .addParams(**"type"**, String.*valueOf*(i)).build().connTimeOut(100000).readTimeOut(100000);  
  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 Log.*i*(**"result"**,**"收到或者丢失------"**+response);  
 **try** {  
 JSONObject json = **new** JSONObject(response);  
 String msg = json.getString(**"msg"**);  
 **int** code = json.getInt(**"code"**);  
 **if** (code == 200) {  
 ToastUtil.*show*(msg);  
 **arrayList**.clear();  
 */// 重新获取数据* message();  
  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
  
 */\*\*  
 \* 2 重新打包 或者 1 不发了  
 \*/* **private void** againPackOrNotSend(**int** code,String orderSn) {  
 RequestCall call = SYApplication.*postFormBuilder*()  
 .url(SYApplication.*path\_url* + **"/user/orders/problemPackage17"**)  
 .addParams(**"order\_sn"**, orderSn)  
 .addParams(**"type"**, String.*valueOf*(code)).build().connTimeOut(100000).readTimeOut(100000);  
  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 Log.*i*(**"result"**,**"重新打包------"**+response);  
 **try** {  
 JSONObject json=**new** JSONObject(response);  
 String msg = json.getString(**"msg"**);  
 **int** code = json.getInt(**"code"**);  
 **if** (code == 200) {  
 ToastUtil.*show*(msg);  
  
 **arrayList**.clear();  
 */// 重新获取数据* message();  
  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
  
 }  
 });  
  
 }  
  
 **private void** message\_error() {  
 RequestCall call = SYApplication.*postFormBuilder*()  
 .url(SYApplication.*path\_url* + **"/user/orders/orders\_deal"**)  
 .addParams(**"order\_sn"**, **order\_sn**)  
 .addParams(**"status"**, **status**)  
 .build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 String msg = jsonObject.getString(**"msg"**);  
 **int** code = jsonObject.getInt(**"code"**);  
 **if** (code == 200) {  
 ToastUtil.*show*(msg);  
 **arrayList**.clear();  
 message();  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 **private void** dialog\_tip(**final int** num) {  
 LayoutInflater inflater\_type = getActivity().getLayoutInflater();  
 View layout\_type = inflater\_type.inflate(R.layout.***dialog\_login***, **null**);  
 AlertDialog.Builder record\_type = **new** AlertDialog.Builder(getActivity());  
 **dialog\_tips** = record\_type.create();  
 **dialog\_tips**.show();  
 **dialog\_tips**.setContentView(layout\_type);  
 **dialog\_tips**.setCanceledOnTouchOutside(**false**);*//* Button btn\_dialog\_sure = layout\_type  
 .findViewById(R.id.***btn\_dialog\_sure***);  
 Button btn\_dialog\_cancel = layout\_type  
 .findViewById(R.id.***btn\_dialog\_cancel***);  
 TextView text\_content = layout\_type  
 .findViewById(R.id.***text\_content***);  
 **if** (num == 1) {  
 text\_content.setText(R.string.***prompt\_tj***);  
 } **else** {  
 text\_content.setText(R.string.***prompt\_delete***);  
 }  
 btn\_dialog\_cancel.setOnClickListener(v -> {  
 *//* ***TODO Auto-generated method stub* dialog\_tips**.dismiss();  
 });  
  
 btn\_dialog\_sure.setOnClickListener(v -> {  
 *//* ***TODO Auto-generated method stub* dialog\_tips**.dismiss();  
 **blockDialog**.show();  
 **if** (num == 1) {  
 message\_tuijian();  
 } **else** {  
 message\_delete();  
 }  
  
  
 });  
 }  
  
 **private void** message\_delete() {  
 RequestCall call = SYApplication.*postFormBuilder*()  
 .url(SYApplication.*path\_url* + **"/user/orders/del"**)  
 .addParams(**"order\_sn"**, **order\_sn**).build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 String msg = jsonObject.getString(**"msg"**);  
 **int** code = jsonObject.getInt(**"code"**);  
 **if** (code == 200) {  
 ToastUtil.*show*(msg);  
 **arrayList**.clear();  
 message();  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 **private void** message\_tuijian() {  
 RequestCall call = SYApplication.*postFormBuilder*()  
 .url(SYApplication.*path\_url* + **"/user/orders/apply\_orders\_back"**)  
 .addParams(**"order\_sn"**, **order\_sn**).build();  
 call.execute(**new** StringCallback() {  
 @Override  
 **public void** onError(Call call, Exception e, **int** id) {  
 **blockDialog**.dismiss();  
 ToastUtil.*show*(e + **""**);  
 }  
  
 @Override  
 **public void** onResponse(String response, **int** id) {  
 **try** {  
 **final** JSONObject jsonObject = **new** JSONObject(response);  
 String msg = jsonObject.getString(**"msg"**);  
 **int** code = jsonObject.getInt(**"code"**);  
 **if** (code == 200) {  
 ToastUtil.*show*(msg);  
 **arrayList**.clear();  
 message();  
 } **else** {  
 ToastUtil.*show*(msg);  
 }  
 } **catch** (JSONException e) {  
 *//* ***TODO Auto-generated catch block*** }  
 }  
 });  
 }  
  
 @Override  
 **protected** Object requestData() {  
 **return** ConstantUtils.***STATE\_SUCCESSED***;  
 }  
@Override  
 **public void** onDestroyView() {  
 **super**.onDestroyView();  
**unbinder**.unbind();  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 StartUtils.*startActivityById*(getActivity(), v.getId());  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 *//* ***TODO: inflate a fragment view*** View rootView = **super**.onCreateView(inflater, container, savedInstanceState);  
 **unbinder** = ButterKnife.*bind*(**this**, rootView);  
 **return** rootView;  
 }  
}

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/all\_bg"  
 android:orientation="vertical"**>  
  
 <**ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:scrollbars="none"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:background="@drawable/login\_bg"  
 android:orientation="vertical"**>  
  
 <**ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="60dp"  
 android:src="@drawable/login\_logo"** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:id="@+id/linear\_login\_content"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="@dimen/height\_ten"  
 android:layout\_marginRight="@dimen/height\_ten"  
 android:layout\_marginTop="140dp"  
 android:background="@color/white"  
 android:orientation="vertical"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_fourty"  
 android:layout\_margin="@dimen/height\_ten"  
 android:background="@drawable/btn\_grey\_top"  
 android:gravity="center\_vertical"  
 android:orientation="horizontal"  
 android:paddingLeft="@dimen/height\_ten"  
 android:paddingRight="@dimen/height\_ten"**>  
  
 <**ImageView  
 android:id="@+id/image\_login\_left"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerVertical="true"  
 android:src="@drawable/login\_account"** />  
  
 <**EditText  
 android:id="@+id/edit\_login\_account"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="@dimen/height\_ten"  
 android:layout\_toRightOf="@+id/image\_login\_left"  
 android:background="@color/transparent"  
 android:hint="@string/login\_account"  
 android:singleLine="true"  
 android:ellipsize="end"  
 android:textColor="@color/main\_black"  
 android:textColorHint="@color/main\_grey9"  
 android:textSize="@dimen/text\_size\_14"** />  
  
 <**LinearLayout  
 android:id="@+id/linear\_login\_type"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentRight="true"  
 android:gravity="center"  
 android:paddingLeft="@dimen/height\_ten"**>  
  
 <**TextView  
 android:id="@+id/text\_login\_type"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="普通用户"  
 android:textColor="@color/main\_black"  
 android:textSize="@dimen/text\_size\_13"** />  
  
 <**ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="@dimen/height\_five"  
 android:src="@drawable/feedback\_bottom"** />  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_fourty"  
 android:layout\_marginLeft="@dimen/height\_ten"  
 android:layout\_marginRight="@dimen/height\_ten"  
 android:background="@drawable/btn\_grey\_top"  
 android:gravity="center\_vertical"  
 android:orientation="horizontal"  
 android:paddingLeft="@dimen/height\_ten"  
 android:paddingRight="@dimen/height\_ten"**>  
  
 <**ImageView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/login\_password"** />  
  
 <**EditText  
 android:id="@+id/edit\_login\_password"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="@dimen/height\_ten"  
 android:background="@color/transparent"  
 android:hint="@string/login\_password"  
 android:singleLine="true"  
 android:ellipsize="end"  
 android:inputType="textPassword"  
 android:textColor="@color/main\_black"  
 android:textColorHint="@color/main\_grey9"  
 android:textSize="@dimen/text\_size\_14"** />  
  
 </**LinearLayout**>  
  
 <**TextView  
 android:id="@+id/relative\_safe\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="40dp"  
 android:layout\_gravity="right"  
 android:layout\_marginRight="@dimen/height\_ten"  
 android:gravity="center\_vertical"  
 android:text="@string/login\_forget\_password"  
 android:textColor="@color/main\_grey9"  
 android:textSize="@dimen/text\_size\_13"** />  
  
 <**Button  
 android:id="@+id/btn\_login\_login"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_fourty"  
 android:layout\_marginBottom="@dimen/height\_twenty"  
 android:layout\_marginLeft="@dimen/height\_ten"  
 android:layout\_marginRight="@dimen/height\_ten"  
 android:layout\_marginTop="@dimen/height\_ten"  
 android:background="@drawable/btn\_red"  
 android:text="@string/login\_login"  
 android:textColor="@color/white"  
 android:textSize="@dimen/text\_size\_14"** />  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/linear\_login\_content"  
 android:orientation="vertical"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="@dimen/height\_thirty"**>  
  
 <**View  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_line"  
 android:layout\_centerVertical="true"  
 android:layout\_marginLeft="@dimen/height\_ten"  
 android:layout\_marginRight="@dimen/height\_twenty"  
 android:layout\_toLeftOf="@+id/text\_login\_quick"  
 android:background="@color/line"** />  
  
 <**TextView  
 android:id="@+id/text\_login\_quick"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:text="@string/login\_third\_login"  
 android:textColor="@color/main\_grey6"  
 android:textSize="@dimen/text\_size\_13"** />  
  
 <**View  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/height\_line"  
 android:layout\_centerVertical="true"  
 android:layout\_marginLeft="@dimen/height\_twenty"  
 android:layout\_marginRight="@dimen/height\_ten"  
 android:layout\_toRightOf="@+id/text\_login\_quick"  
 android:background="@color/line"** />  
 </**RelativeLayout**>  
  
  
 <**LinearLayout  
 android:id="@+id/linear\_login\_facebook"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="@dimen/height\_ten"  
 android:gravity="center"  
 android:orientation="vertical"**>  
  
 <**ImageView  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:src="@drawable/login\_weixin"** />  
  
 </**LinearLayout**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_margin="@dimen/height\_twenty"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/login\_account\_no"  
 android:textColor="@color/main\_grey6"  
 android:textSize="@dimen/text\_size\_13"** />  
  
 <**TextView  
 android:id="@+id/text\_login\_register"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/login\_register"  
 android:textColor="@color/orange"  
 android:textSize="@dimen/text\_size\_13"** />  
  
 </**LinearLayout**>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 </**ScrollView**>  
</**RelativeLayout**>