

ContactDetailActivity

```
initView this.titleName = (TextView)this.findViewById(id.titlename); this.titleLeftButton = (ImageButton)this.findViewById(id.titleleftbutton);
this.titleLeftButton.setVisibility(0); this.titleLeftButton.setOnClickListener(new OnClickListener() { public void onClick(View var1) {
ContactDetailActivity.this.finish(); } }); this.OP = (TextView)this.findViewById(id.personname); this.aeT =
(TextView)this.findViewById(id.confirmdadddelete); this.avatar = (ImageView)this.findViewById(id.avatar); this.aeU =
(LinearLayout)this.findViewById(id.mxpersioncircle); this.aeV = (LinearLayout)this.findViewById(id.mxpersionfiles); this.aeW =
(Button)this.findViewById(id.sendsms); this.firstloading = (ProgressBar)this.findViewById(id.firstloading); this.aeX =
(RelativeLayout)this.findViewById(id.persondetailcontianer); this.aeY = (Switch)this.findViewById(id.mxswithattention); this.aeZ =
(LinearLayout)this.findViewById(id.sendlayout); this.afa = (LinearLayout)this.findViewById(id.addordelete_layout); this.afb =
(RelativeLayout)this.findViewById(id.mxspacialattention); u.a(this.aeX, w.au(this).density);
```

```
this.zP = new b();
```

```
ContactDetailActivity.this.aeP = (WBPersonExtension)var1;
```

```
this.personID = this.getIntent().getIntentExtra("person_id", -999);
```

```
UserAccount var2 = df.iA().iB(); if (var2 != null && var2.getCurrentIdentity() != null) { this.currentUserID = var2.getCurrentIdentity().getId(); }
```

添加联系人: AddPersonalContactActivity.java

```
onCreate() this.initView(); this.zP = new b(); this.listView.setOnItemClickListener(new OnItemClickListener() { public void onItemClick(AdapterView<?
> var1, View var2, int var3, long var4) { ContactQueryByNameActivity.this.d((ContactPeople)ContactQueryByNameActivity.this.ags.get(var3)); } });
```

```
initView() RelativeLayout var1 = (RelativeLayout)LayoutInflater.from(this).inflate(layout.mxcontactsearch_main, (ViewGroup)null);
this.setContentview(var1);
```

```
TextView var2 = (TextView)this.findViewById(id.title_name);
var2.setText(string.mx_top_right_add_contact);

this.aew = (RelativeLayout)this.findViewById(id.search_by_name); 按姓名查找
this.aex = (RelativeLayout)this.findViewById(id.find_more_star_friend); 添加公众号
this.aey = (RelativeLayout)this.findViewById(id.search_contacts);公司通讯录
this.aez = (RelativeLayout)this.findViewById(id.search_add_new_friend);添加新好友
this.aeE = this.findViewById(id.view_mx_contact_search_main_friend_divider); (只是一条分割线)
```

```
mxcontactsearchmain.xml <include android:id="@+id/systemtitle" layout="@layout/mxsystemtitle" />
```

```
mxsystemtitle.xml id.title_name
```

```
按姓名查找点击事件: this.aew.setOnClickListener(new OnClickListener() { public void onClick(View var1) { Intent var2 = new
Intent(AddPersonalContactActivity.this, ContactQueryByNameActivity.class); AddPersonalContactActivity.this.startActivity(var2); } });
```

搜姓名 ContactQueryByNameActivity.java

```
private List<IContact> ags = new ArrayList(); //ListView中item展示的内容集合
private ei agu = null; //继承自BaseAdapter的ListView的适配器
```

```
initView() this.setContentview(layout.mxcontactsearchname); this.listView = (ListView)this.findViewById(id.contactlist); this.zF =
(EditText)this.findViewById(id.searchName); 请输入姓名或拼音 this.zH = (ImageView)this.findViewById(id.removeicon); 清除EditText的图标
this.leftbutton = (ImageButton)this.findViewById(id.titleleftbutton); 退出此Activity的返回图标 this.agr = (Button)this.findViewById(id.titleright_button);
查找
```

```
this.system_titleName = (TextView)this.findViewById(id.title_name);
this.system_titleName.setText(string.mx_search_name); 搜姓名
```

```
mxcontactsearchname.xml <include android:id="@+id/systemtitle" layout="@layout/mxsystemtitle" />
```

```
mxsystemtitle.xml id.title_name id.title/leftbutton
```

查找的点击事件 this.agr.setOnClickListener(new OnClickListener() { public void onClick(View var1) { ContactQueryByNameActivity.this.mb(); } });

this.mb()方法实现: private void mb() { String var1 = this.zF.getText().toString(); //var1表示EditText中输入的姓名关键字 if (var1 != null && var1.trim().length() != 0) { //r 是MXSDK中的工具类, cd 方法对var1字符串做过滤, 不能是数字开头的纯数字, //当var1符合正则表达式"^d+\$"返回 true if (r.cb(var1)) { //u 是MXSDK中的工具类, b 方法用于弹出Toast提示信息 "请输入要查询的姓名" u.b(this, this.getString(string.mxtoastpleaseinputnametosearch), 0); } else { this.cN(var1); //此方法真正实现查询功能 } } else { this.ags.clear(); //清除ListView this.prepareViewData(); this.nodata.setVisibility(8); //8表示GONE }

```
}
```

prepareViewData() 方法实现 用于初始化ListView, 并设置它的适配器 agu private void prepareViewData() { this.agu = new ei(this.ags, this, 100, 200, false, 0, (List)null); this.listView.setAdapter(this.agu); this.agu.notifyDataSetChanged(); }

this.cN(String var1) 方法的实现 //此方法真正实现查询功能 private void cN(String var1) { //var1表示要查询的姓名关键字 this.ags.clear(); this.agr.setEnabled(false); this.zP.e(var1, new n(this) { //this.zP是 MXSDK中 b 类 的一个对象 public void success(Object var1) { ContactQueryByNameActivity.this.firstloading.setVisibility(8); List var2 = (List)var1; ContactQueryByNameActivity.this.ags.addAll(var2); if (ContactQueryByNameActivity.this.ags.isEmpty()) { ContactQueryByNameActivity.this.nodata.setVisibility(0); } else { ContactQueryByNameActivity.this.nodata.setVisibility(8); }

```
        ContactQueryByNameActivity.this.prepareViewData();
        ContactQueryByNameActivity.this.agr.setEnabled(true);
    }

    public void failure(MXError var1) {
        super.failure(var1);
        ContactQueryByNameActivity.this.agr.setEnabled(true);
    }
});
}
```

MXSDK中的 b.java

b 类是一个服务类（不是继承自Service的服务类），处理 HTTP请求的入口 package com.minxing.kit.internal.core.service; public class b

```
private eu alU = new eu() {
    public void success(Object var1) {
        dr var2 = new dr();
        this.mCallBack.success(var2.a((JSONObject)var1, -999));
    }

    public void failure(MXError var1) {
        this.mCallBack.failure(var1);
    }
};

public void e(String var1, n var2) {
    d var3 = new d();
    var3.a(MXMethod.GET);
    ArrayList var4 = new ArrayList();
    if (var1 != null && !"".equals(var1)) {
        var4.add(new BasicNameValuePair("q", var1));
    }

    var4.add(new BasicNameValuePair("limit", "50"));
    var3.setHeaders((TreeMap)null);
    var3.H((List)null);
    var3.setParams(var4);
    var3.a(MXInterface.CONTACT_SEARCH);
    this.alU.setViewCallBack(var2);
    (new ew(this.alU)).b(var3);
}
```

MXSDK中的 a.java

```
package com.minxing.kit.internal.core;
public abstract class a

public a(Context var1, boolean var2, String var3, String var4) {
    //参数var2, var3, var4跟弹出提示对话框相关, 跟事件回调无关, 无需关注
    this.mContext = var1;
    this.dialogContent = var4;
    this.dialogTitle = var3;
    this.isShowProgressDialog = var2;
}

public abstract void success(Object var1); //请求成功的回调,

public abstract void failure(MXError var1); //请求失败的回调
```

MXSDK中的 n.java

n 类只是一个监听回调类, 通过重写它的success(Object var1) 方法 处理请求成功的返回结果 重写failure(MXError var1) 方法处理请求失败的返回结果 package com.minxing.kit.internal.core.service public class n extends a

MXSDK中的 eu.java

eu 类是一个抽象类, 继承自 a类, 所以 eu类 也是一个 监听回调类, 此类的目的是包装, 它可以把回调的结果做一次处理后, 得到处理后的结果, 再传递给 mCallBack监听回调类中的回调方法。 参考MXSDK b 类中的 alU 成员变量

```
package com.minxing.colorpicker;
public abstract class eu extends a

public a mCallBack;

public void setViewCallBack(a var1) {
    if (var1 != null) {
        this.mCallBack = var1;
        this.mContext = var1.getContext();
        this.isShowProgressDialog = var1.isShowProgressDialog();
        this.dialogTitle = var1.getDialogTitle();
        this.dialogContent = var1.getDialogContent();
    }
}
```

MXSDK中的 d.java

d 类用于封装请求参数 package com.minxing.kit.internal.core public class d

```
private MXMethod ajV; //ajV表示请求方式
private MXInterface ajW; //ajW表示请求类型（本次请求的目的是什么，如查找联系人）
private TreeMap<String, String> headers;
private List<NameValuePair> params;
private List<UploadFile> ajX; //ajX表示要上传的文件集合

//aka字符串用来表示请求地址url的部分内容，可能是path部分，
//如果aka为null，那么请求地址则使用MXInterface中的formatFace字符串
private String aka;

public MXMethod nf() { //nf 方法用于返回 d对象中封装的请求方式
    return this.ajV;
}

public void a(MXMethod var1) { //a(MXMethod var1) 方法用于 设置 d对象中的请求方式
    this.ajV = var1;
}

public MXInterface ng() { //nf 方法用于返回 d对象中封装的请求类型
```

```

        return this.ajW;
    }

    public void a(MXInterface var1) { //a(MXInterface var1) 方法用于 设置 d对象中的请求类型
        this.ajW = var1;
    }

    public List<UploadFile> nh() { //nh 方法返回 d对象中封装的要上传的文件集合
        return this.ajX;
    }

    public void H(List<UploadFile> var1) { //H 方法设置 d对象中封装的要上传的文件集合
        this.ajX = var1;
    }

    public TreeMap<String, String> getHeaders() {
        return this.headers;
    }

    public void setHeaders(TreeMap<String, String> var1) { 设置 d对象中的 请求头信息
        this.headers = var1;
    }

    public List<NameValuePair> getParams() {
        return this.params;
    }

    public void setParams(List<NameValuePair> var1) { 设置 d对象的 请求参数集合 params，可用于url中的查询条件
        this.params = var1;
    }

    public void cT(String var1) {
        this.aka = var1;
    }

    public String nk() {
        return this.aka;
    }
}

```

MXSDK中的 MXMethod.java

MXMethod是一个枚举类，封装请求方式，如GET、POST等 package com.minxing.kit.internal.core public enum MXMethod

```

GET("GET"),
POST("POST"),
PUT("PUT"),
DELETE("DELETE");

private String mMethod;

private MXMethod(String var3) {
    this.mMethod = var3;
}

public String getMethod() {
    return this.mMethod;
}

```

MXSDK中的 ew.java

ew 类继承自AsyncTask，表示ew类 是用于在后台处理线程任务，并返回结果 package com.minxing.colorpicker; //d类型表示执行后台任务时传入的参数Params, Object类型表示返回的结果类型 public class ew extends AsyncTask<d, Void, Object>

```

private static ExecutorService aka = Executors.newScheduledThreadPool(5); //ew类中自定义了一个线程池 aka

public ew(a var1) { //构造方法中传入的var1一个监听回调对象

```

```

        this.mCallBack = var1;
        this.mContext = var1.getContext();
    }

    public void b(d var1) {
        if (VERSION.SDK_INT >= 11) {
            //Android3.0 之后使用自定义的线程池 akA 执行线程任务，避免 AsyncTask默认的串行执行
            this.executeOnExecutor(akA, new d[]{var1});
        } else {
            super.execute(new d[]{var1});
        }
    }

    // 此方法就是 AysncTask的 doInBackground方法
    public Object a(d... var1) {
        d var2 = var1[0];
        if (var2 == null) {
            return var2;
        } else {
            String var3 = null;
            if (var2.nk() == null) {
                MXInterface var4 = var2.ng();
                var3 = var4.getFormatFace();
            } else {
                var3 = var2.nk();
            }

            return var2.nj() ? this.c(var2) : this.d(var2);
        }
    }

    private Object d(d var1) {
        return this.a(new ev(), var1);
    }

    private Object a(ev var1, d var2) {
        Object var3 = null;

        ...

        var3 = var1.a(this.mContext, var2);

        ...

        return var3;
    }
}

```

MXSDK中的 ev.java

ev 类中 通过HttpClient框架发送 HTTP请求 package com.minxing.colorpicker; public class ev

```

    public Object a(Context var1, d var2) throws Exception {
        ...
        String var3 = this.a(var2);
        Class var4 = var2.ng().getClazz();
        ...

        else {
            try {
                //通过fastJsong框架解析Json字符串 var3，其中var4来自MXInterface，通常就是一个Object
                Object var6 = JSON.parseObject(var3, var4);
                return var6;
            } catch (Exception var15) {
                ...
            }
        }
    }
}

```

//此方法正在处理HTTP请求

```
protected String a(d var1) throws Exception {  
    MXMethod var2 = var1.nf();  
    TreeMap var3 = var1.getHeaders();  
    List var4 = var1.getParams();  
    List var5 = var1.nh();  
}
```