

Android中能够简化开发流程的一些框架

本文介绍的是一些博主在开发过程中经常用到的Android开源框架，所谓开源框架我的理解就是别人封装好的代码，可以直接拿过来使用，并且源码也全部公开的代码库。我对于开源框架的使用的态度是，如果完全符合我们项目的需求，或者可定制化的程度非常高的话，那么便可以拿过来直接用，因为开源框架的源码都在那里，如果遇到和项目预期不一样的地方我们也可以把源码拿过来自己改一下，然后重新打个包嘛。但是，不是很建议刚开始学习安卓的小伙伴们，只会用第三方框架，但是不理解框架内部的实现，在用第三方框架的过程中，我们是可以自己先封装一下简单的框架的，然后再了解一下别人框架内部实现的逻辑什么样的，知其所以然嘛。同时也非常感谢这些开源框架的作者们，他们的开源精神，真的是非常的伟大啊。

网络框架

- [NoHttp](#)
- [RxJava](#) + [retrofit](#)
- [Okhttp](#)
- [okhttp-OkGo](#)

Json解析框架

- [fastJson](#)
- [gson](#)

图片加载框架

- [fresco](#)
- [Glide](#)
- [picasso](#)

Log类的库

- [logger](#)
- [KLog](#)
- [xLog](#)

RecyclerView

- [UltimateRecyclerView](#)
- [IndexRecyclerView](#)
- [recyclerview-animators](#)

Adapter

- [BaseRecyclerViewAdapterHelper](#)
- [baseAdapter](#)
- [base-adapter-helper](#)

数据库

- [greenDAO](#)
- [realm-java](#)
- [ormlite-android](#)

注解库

- [butterknife](#)
- [xUtils3](#)
- [annotations](#)

事件总线

- [EventBus](#)
- [RxJava](#)

图片剪裁

- [uCrop](#)
- [android-crop](#)
- [glide-transformations](#)

性能检测

- [leakcanary](#)

后台任务队列

- [tape](#)

UI

- [Android-Bootstrap](#)
- [material-dialogs](#)
- [Android-PickerView](#)
- [TastyToast](#)
- [awesome-android-ui](#)

感觉这么给大家介绍完了，可能大家会感觉到很抽象，所以打算动手撸一个小的项目，让大家具体感受一下大神们封装的库

[项目源码下载](#)

项目中用到的库：

- nohttp
- butterknife
- glide
- logger

- BaseRecyclerViewAdapterHelper
- eventbus
- glide-transformations
- leakcanary
- Android-Bootstrap
- TastyToast
- material-dialogs

项目截图:

nohttp:

```

//同步请求,结果直接存储在了response
Request<String> request = NoHttp.createStringRequest("自己的url", RequestMethod.POST);
Response<String> response = NoHttp.startRequestSync(request);
String result = response.get();
//可以直接将结果取出,并且内置了Gson,fastJson,可以直接将结果转换成对应的model

//异步请求,可以构建请求队列,默认同时三个请求一起,参数可调
RequestQueue requestQueue = NoHttp.newRequestQueue();

OnResponseListener<String> listener = new OnResponseListener<String>() {
    @Override public void onStart(int what) {

    }

    @Override public void onSuccess(int what, Response response) {

    }

    @Override public void onFailed(int what, Response response) {

    }

    @Override public void onFinish(int what) {

    }
};

//请求String
Request<String> request2 = NoHttp.createStringRequest("自己的url", RequestMethod.GET);
requestQueue.add(0, request2, listener);

//可以自定义请求类型

/*
// JsonObject
Request<JSONObject> objRequest = NoHttp.createJsonObjectRequest("自己的url", RequestMethod.POST);
requestQueue.add(0, objRequest, listener);

// JSONArray
Request<JSONArray> arrayRequest = NoHttp.createJSONArrayRequest("自己的url", RequestMethod.PUT);
requestQueue.add(0, arrayRequest, listener);

Request<JSONObject> request = new FastJsonRequest(url, RequestMethod.POST);
requestQueue.add(0, request, listener);

// 内部使用Gson、FastJson解析成JavaBean
Request<UserInfo> request = new JavaBeanRequest(url, RequestMethod.GET);
requestQueue.add(0, request, listener);

Request<JSONObject> request = new JavaBeanRequest(url, RequestMethod.POST);
.add("name", "yoldada") // String类型
.add("age", 18) // int类型
.add("sex", '0') // char类型
.add("time", 16346468473154) // long类型

// 添加Bitmap
.add("head", new BitmapBinary(bitmap))
// 添加File
.add("head", new FileBinary(file))
// 添加ByteArray
.add("head", new ByteArrayBinary(byte[]))
// 添加InputStream
.add("head", new InputStreamBinary(inputStream));

*/

//nohttp同时有非常完备的缓存的策略,同时对文件上传,请求包体都有非常好的兼容,并且全都是中文文档~

```

butterknife:

```

@BindView(R.id.title) TextView title;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ButterKnife.bind(this);
    title.setText("一些第三方库的测试的demo");

}

@OnClick(R.id.nohttp) public void onNohttpClicked() {
    startActivity(new Intent(MainActivity.this, NoHttpActivity.class));
}

@OnClick(R.id.glide) public void onGlideClicked() {
    startActivity(new Intent(MainActivity.this, GlideActivity.class));
}

@OnClick(R.id.baseRecyclerview_adapter_helper) public void onBaseRecyclerviewAdapterHelperClicked() {
    startActivity(new Intent(MainActivity.this, BaseRecyclerViewAdapterActivity.class));
}

@OnClick(R.id.event_bus) public void onEventBusClicked() {
    startActivity(new Intent(MainActivity.this, EventBusActivity.class));
}

@OnClick(R.id.glide_transformations) public void onGlideTransformationsClicked() {
    startActivity(new Intent(MainActivity.this, GlideTransformationActivity.class));
}

@OnClick(R.id.tasty_toast) public void onTastyToastClicked() {
    startActivity(new Intent(MainActivity.this, TastyToastActivity.class));
}

@OnClick(R.id.material_dialogs) public void onMaterialDialogsClicked() {
    startActivity(new Intent(MainActivity.this, MaterialDialogActivity.class));
}

@OnClick(R.id.logger) public void onLoggerClicked() {
    startActivity(new Intent(MainActivity.this, LoggerActivity.class));
}

```

glide:

```

Glide.with(this).load("http://goo.gl/gEgYUd").into(img1);

Glide.with(this)
    .load("http://goo.gl/gEgYUd")
    .fitCenter()
    .centerCrop()
    .into(img2);

Glide.with(this)
    .load("")
    .fitCenter()
    .centerCrop()
    .error(R.mipmap.ic_launcher)
    .into(img3);

Glide.with(this)
    .load("http://goo.gl/gEgYUd")
    .diskCacheStrategy(DiskCacheStrategy.ALL)
    .into(img4);

Glide.with(this)
    .load("http://goo.gl/gEgYUd")
    .crossFade()
    .into(img5);

Glide.with(this)
    .load("http://goo.gl/gEgYUd")
    .into(new SimpleTarget<GlideDrawable>() {
        @Override public void onResourceReady(GlideDrawable resource, GlideAnimation<?> su
            img6.setImageDrawable(resource);
        }
    });

```

logger:

```

Logger.d("hello");
Logger.d("hello %s %d", "world", 5); // String.format
Logger.d("hello");
Logger.e("hello");
Logger.w("hello");
Logger.v("hello");
Logger.wtf("hello");

//      Logger.json(JSON_CONTENT); //打印json
//      Logger.xml(XML_CONTENT);
//      Logger.log(DEBUG, "tag", "message", throwable);
//      Logger.d("hello %s", "world");
//
//      Logger.d(list); //打印list
//      Logger.d(map);
//      Logger.d(set);
//      Logger.d(new String[]);
//
//      Logger
//          .init(YOUR_TAG) // default PRETTYLOGGER or use just init()
//          .methodCount(3) // default 2
//          .hideThreadInfo() // default shown
//          .logLevel(LogLevel.NONE) // default LogLevel.FULL
//          .methodOffset(2) // default 0
//          .logAdapter(new AndroidLogAdapter()); //default AndroidLogAdapter
//
//
//
Logger.log(5000, "tag", "内容", null); //延时打log

```

BaseRecyclerViewAdapterHelper:

```

MultipleItem item1 = new MultipleItem(1, "aaa");
MultipleItem item2 = new MultipleItem(1, "bbb");
MultipleItem item3 = new MultipleItem(1, "ccc");
MultipleItem item4 = new MultipleItem(1, "ddd");

```

```

MultipleItem item5 = new MultipleItem(2, "");
MultipleItem item6 = new MultipleItem(2, "");

List<MultipleItem> list = new ArrayList<MultipleItem>();

list.add(item1);
list.add(item2);
list.add(item5);
list.add(item3);
list.add(item4);
list.add(item6);

MultipleItemQuickAdapter adapter = new MultipleItemQuickAdapter(list);
adapter.setSpanSizeLookup(new BaseQuickAdapter.SpanSizeLookup() {
    @Override public int getSpanSize(GridLayoutManager gridLayoutManager, int position) {
        if (position == 2 || position == 5) {
            return 4;
        } else {
            return 2;
        }
    }
});
recyclerView.setLayoutManager(new GridLayoutManager(this, 4));
recyclerView.setAdapter(adapter);

// 轻松实现多种布局，点击事件等
// 添加动画等

adapter.setOnItemClickListener(new BaseQuickAdapter.OnItemClickListener() { // item点击事件
    @Override public void onItemClick(BaseQuickAdapter adapter, View view, int position) {
        Toast.makeText(BaseRecyclerViewAdapterActivity.this, "点击了" + position, Toast.LENGTH
    }
});

// adapter.addHeaderView(); 添加headerview

// 可以优化Adapter代码
// 添加Item事件
// 添加列表加载动画
// 添加头部、尾部
// 自动加载
// 添加分组
// 自定义不同的item类型
// 设置空布局
// 添加拖拽、滑动删除
// 分组的伸缩栏
// 自定义ViewHolder
}

public class MultipleItem implements MultiItemEntity {
    public static final int TEXT = 1;
    public static final int IMG = 2;
    public String text;
    private int itemType;

    public MultipleItem(int itemType, String text) {
        this.itemType = itemType;
        this.text = text;
    }

    @Override
    public int getItemType() {
        return itemType;
    }
}

public class MultipleItemQuickAdapter extends BaseMultiItemQuickAdapter<MultipleItem, BaseViewHol

    public MultipleItemQuickAdapter(List data) {
        super(data);
        addItemType(MultipleItem.TEXT, R.layout.item_recycler_view);
        addItemType(MultipleItem.IMG, R.layout.item_recycler_view2);
    }

    @Override
    protected void convert(BaseViewHolder helper, MultipleItem item) {
        switch (helper.getItemViewType()) {
            case MultipleItem.TEXT:

```

```

        helper.setText(R.id.text, item.text);
        break;
    case MultipleItem.IMG:
        // 可以在这里设置图片, 现在默认是背景图片
        break;
    }
}
}

```

eventbus:

```

// 当然, 我们目前将, 发送与接收放在了一个界面里面, 他们可以在不同的界面里面, 可以实现app内部的通信

@Override protected void onStart() {
    super.onStart();
    EventBus.getDefault().register(this);
}

@Override protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_eventbus);
    ButterKnife.bind(this);
}

@Subscribe(threadMode = ThreadMode.MAIN)
public void onMessageEvent(MessageEvent event) {
    Toast.makeText(EventBusActivity.this, "收到的内容 -> " + event.event, Toast.LENGTH_SHORT).show();
}

@Override protected void onDestroy() {
    EventBus.getDefault().unregister(this);
    super.onDestroy();
}

@OnClick(R.id.send) public void onViewClicked() {
    EventBus.getDefault().post(new MessageEvent("今天好开心"));
}

public class MessageEvent {
    String event;

    public MessageEvent(String event) {
        this.event = event;
    }
}

```

glide-transformations:

```

// 可以修改颜色, 加滤镜, 剪裁, gpu加速等

Glide.with(this).load("https://ws1.sinaimg.cn/large/006tNbRwly1ffs6l68mx3j30ge0gen08.jpg")
    .bitmapTransform(new BlurTransformation(this, 30), new CropCircleTransformation(this))
    .into((ImageView) findViewById(R.id.img1));

Glide.with(this).load("https://ws1.sinaimg.cn/large/006tNbRwly1ffs6l68mx3j30ge0gen08.jpg")
    .bitmapTransform(new CropCircleTransformation(this))
    .into((ImageView) findViewById(R.id.img2));

Glide.with(this).load("https://ws1.sinaimg.cn/large/006tNbRwly1ffs6l68mx3j30ge0gen08.jpg")
    .bitmapTransform(new CropSquareTransformation(this))
    .into((ImageView) findViewById(R.id.img3));

Glide.with(this).load("https://ws1.sinaimg.cn/large/006tNbRwly1ffs6l68mx3j30ge0gen08.jpg")
    .bitmapTransform(new ColorFilterTransformation(this, 0x80dfdfdf))
    .into((ImageView) findViewById(R.id.img4));

```


leakcanary:

```
/*
 * LeakCanary用来专注分析系统的进程
 */

if (LeakCanary.isInAnalyzerProcess(this)) {
    return;
}
```

Android-Bootstrap:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="12dp"
    android:layout_marginRight="12dp"
    android:orientation="vertical"
    tools:context="com.dotengine.linsir.basedevelop.MainActivity">

    <TextView
        android:id="@+id/title"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:gravity="center"
        android:text="常用库的demo"
        android:textSize="20sp"/>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="#dfdfdf"/>

    <com.beardedhen.androidbootstrap.BootstrapButton
        android:id="@+id/nohttp"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:text="nohttp"
        app:bootstrapBrand="success"
        app:bootstrapSize="lg"
        app:buttonMode="regular"
        app:roundedCorners="true"
        app:showOutline="false"
        android:layout_marginTop="12dp"
        />

    <com.beardedhen.androidbootstrap.BootstrapButton
        android:id="@+id/glide"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:text="glide"
        android:layout_marginTop="12dp"
        app:bootstrapBrand="warning"
        app:bootstrapSize="lg"
        app:buttonMode="regular"
        app:roundedCorners="true"
        app:showOutline="false"
        />

    <com.beardedhen.androidbootstrap.BootstrapButton
        android:id="@+id/baseRecyclerView_adapter_helper"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:text="baseRecyclerView-Adapter-Helper"
        android:layout_marginTop="12dp"
        app:bootstrapBrand="primary"
        app:bootstrapSize="lg"
        app:buttonMode="regular"
        app:roundedCorners="true"
        app:showOutline="false"
        />
```

```
<com.beardedhen.androidbootstrap.BootstrapButton
    android:id="@+id/event_bus"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:text="eventbus"
    android:layout_marginTop="12dp"
    app:bootstrapBrand="danger"
    app:bootstrapSize="lg"
    app:buttonMode="regular"
    app:roundedCorners="true"
    app:showOutline="false"
/>
```

```
<com.beardedhen.androidbootstrap.BootstrapButton
    android:id="@+id/glide_transformations"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginTop="12dp"
    android:text="glide-transformations"
    app:bootstrapBrand="success"
    app:bootstrapSize="lg"
    app:buttonMode="regular"
    app:roundedCorners="true"
    app:showOutline="true"/>
```

```
<com.beardedhen.androidbootstrap.BootstrapButton
    android:id="@+id/tasty_toast"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:text="TastyToast"
    android:layout_marginTop="12dp"
    app:bootstrapBrand="warning"
    app:bootstrapSize="lg"
    app:buttonMode="regular"
    app:roundedCorners="true"
    app:showOutline="true"
/>
```

```
<com.beardedhen.androidbootstrap.BootstrapButton
    android:id="@+id/material_dialogs"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:text="material-dialogs"
    android:layout_marginTop="12dp"
    app:bootstrapBrand="primary"
    app:bootstrapSize="lg"
    app:buttonMode="regular"
    app:roundedCorners="true"
    app:showOutline="true"
/>
```

```
<com.beardedhen.androidbootstrap.BootstrapButton
    android:id="@+id/logger"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:text="logger"
    android:layout_marginTop="12dp"
    app:bootstrapBrand="danger"
    app:bootstrapSize="lg"
    app:buttonMode="regular"
    app:roundedCorners="true"
    app:showOutline="true"
/>
```

```
</LinearLayout>
```



TastyToast:

```

@OnClick(R.id.button1) public void onButton1Clicked() {
    TastyToast.makeText(getApplicationContext(), "Hello World !", TastyToast.LENGTH_LONG, TastyTo
}

@OnClick(R.id.button2) public void onButton2Clicked() {
    TastyToast.makeText(getApplicationContext(), "Hello World !", TastyToast.LENGTH_LONG, TastyTo
}

@OnClick(R.id.button3) public void onButton3Clicked() {
    TastyToast.makeText(getApplicationContext(), "Hello World !", TastyToast.LENGTH_LONG, TastyTo
}

@OnClick(R.id.button4) public void onButton4Clicked() {
    TastyToast.makeText(getApplicationContext(), "Hello World !", TastyToast.LENGTH_LONG, TastyTo
}

@OnClick(R.id.button5) public void onButton5Clicked() {
    TastyToast.makeText(getApplicationContext(), "Hello World !", TastyToast.LENGTH_LONG, TastyTo
}

@OnClick(R.id.button6) public void onButton6Clicked() {
    TastyToast.makeText(getApplicationContext(), "Hello World !", TastyToast.LENGTH_LONG, TastyTo
}

```

material-dialogs:

```

@OnClick(R.id.button1) public void onButton1Clicked() {
    new MaterialDialog.Builder(this)
        .title("标题")
        .content("是否同意")
        .positiveText("同意")
        .onPositive(new MaterialDialog.SingleButtonCallback() {
            @Override public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogActi
                Toast.makeText(MaterialDialogActivity.this, "同意", Toast.LENGTH_SHORT).show()
        })
        .negativeText("不同意")
        .onNegative(new MaterialDialog.SingleButtonCallback() {
            @Override public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogActi
                Toast.makeText(MaterialDialogActivity.this, "不同意", Toast.LENGTH_SHORT).show()
        })
        .show();
}

@OnClick(R.id.button2) public void onButton2Clicked() {
    MaterialDialog.Builder builder = new MaterialDialog.Builder(this)
        .title("标题")
        .content("是否同意")
        .positiveText("同意");

    MaterialDialog dialog = builder.build();
    dialog.show();
    //dialog.dismiss();
}

@OnClick(R.id.button3) public void onButton3Clicked() {
    new MaterialDialog.Builder(this)
        .title("标题")
        .content("是否同意")
        .positiveText("1111")
        .negativeText("2222")
        .neutralText("3333")
        .show();
}

@OnClick(R.id.button4) public void onButton4Clicked() {
    new MaterialDialog.Builder(this)
        .onPositive(new MaterialDialog.SingleButtonCallback() {
            @Override
                public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which)
                    Toast.makeText(MaterialDialogActivity.this, "1111", Toast.LENGTH_SHORT).show()
        })
}

```

```

        .onNeutral(new MaterialDialog.SingleButtonCallback() {
            @Override
            public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which) {
                Toast.makeText(MaterialDialogActivity.this, "2222", Toast.LENGTH_SHORT).show();
            }
        })
        .onNegative(new MaterialDialog.SingleButtonCallback() {
            @Override
            public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which) {
                Toast.makeText(MaterialDialogActivity.this, "3333", Toast.LENGTH_SHORT).show();
            }
        })
        .onAny(new MaterialDialog.SingleButtonCallback() {
            @Override
            public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which) {
                Toast.makeText(MaterialDialogActivity.this, "4444", Toast.LENGTH_SHORT).show();
            }
        });

    @OnClick(R.id.button5) public void onButton5Clicked() {
        new MaterialDialog.Builder(this)
            .iconRes(R.mipmap.ic_launcher)
            .limitIconToDefaultSize()
            .title("test")
            .positiveText("allow")
            .negativeText("deny")
            .onAny(new MaterialDialog.SingleButtonCallback() {
                @Override
                public void onClick(@NonNull MaterialDialog dialog, @NonNull DialogAction which) {
                    Toast.makeText(MaterialDialogActivity.this, "Prompt checked? " + dialog.isPro
                }
            })
            .checkBoxPromptRes(R.string.app_name, false, null)
            .show();
    }

    @OnClick(R.id.button6) public void onButton6Clicked() {
        new MaterialDialog.Builder(this)
            .title("test")
            .items(R.array.good)
            .itemsCallback(new MaterialDialog.ListCallback() {
                @Override
                public void onSelection(MaterialDialog dialog, View view, int which, CharSequence
                    Toast.makeText(MaterialDialogActivity.this, "点击的是 " + which, Toast.LENGTH_
                }
            })
            .show();
    }

    @OnClick(R.id.button7) public void onButton7Clicked() {
        new MaterialDialog.Builder(this)
            .title("test")
            .items(R.array.good)
            .itemsCallbackSingleChoice(-1, new MaterialDialog.ListCallbackSingleChoice() {
                @Override
                public boolean onSelection(MaterialDialog dialog, View view, int which, CharSeque
                    /**
                     * If you use alwaysCallSingleChoiceCallback(), which is discussed below,
                     * returning false here won't allow the newly selected radio button to actual
                     */
                    return true;
                }
            })
            .positiveText(R.string.app_name)
            .show();
    }

```

[项目源码下载](#)

以上便是对这些库的一个小的总结，大家如果感觉这些库不错的话，建议上github上面看一下他们的文档，可以学到更多的用法，并且可以看到他们的源码。希望大家同样可以上github给我star，或者follow~ [项目源码下载](#)

