Android快速开发手册

# Dependencies

// recyclerview

implementation 'com.android.support:recyclerview-v7:27.1.0'

//butterknife

implementation 'com.jakewharton:butterknife:8.8.1'

annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'

//rxjava2+retrofit

compile **'io.reactivex.rxjava2:rxandroid:2.0.1'**compile **'io.reactivex.rxjava2:rxjava:2.1.7'**compile **'com.squareup.retrofit2:retrofit:2.3.0'**compile **'com.squareup.retrofit2:converter-gson:2.3.0'**compile **'com.squareup.retrofit2:adapter-rxjava2:2.3.0'**compile **'com.squareup.picasso:picasso:2.5.2'**

# ViewPager+tabLayout+Navigation Drawer

1. **创建xml**

<**android.support.v4.widget.DrawerLayout  
 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:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".tasks.TasksActivity"  
 tools:openDrawer="start"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="?attr/colorPrimary"  
 android:minHeight="?attr/actionBarSize"  
 android:theme="@style/Toolbar"  
 app:popupTheme="@style/ThemeOverlay.AppCompat.Light"** />  
 </**android.support.design.widget.AppBarLayout**>  
  
 <**android.support.design.widget.CoordinatorLayout  
 android:id="@+id/coordinatorLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**FrameLayout  
 android:id="@+id/contentFrame"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
 <**android.support.design.widget.FloatingActionButton  
 android:id="@+id/fab\_add\_task"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="@dimen/fab\_margin"  
 android:src="@drawable/ic\_add"  
 app:layout\_anchor="@id/contentFrame"  
 app:layout\_anchorGravity="bottom|right|end"  
 app:fabSize="normal"** />  
 </**android.support.design.widget.CoordinatorLayout**>  
  
 </**LinearLayout**>  
  
 *<!-- Navigation Drawer -->* <**android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header"  
 app:menu="@menu/drawer\_actions"** />  
  
  
</**android.support.v4.widget.DrawerLayout**>

1. **更新Activity文件**

**public class** TasksActivity **extends** AppCompatActivity {  
  
 **private** DrawerLayout **mDrawerLayout**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***tasks\_act***);  
 ButterKnife.*bind*(**this**);  
  
  
 *// Set up the toolbar.* Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
 ActionBar ab = getSupportActionBar();  
 ab.setHomeAsUpIndicator(R.drawable.***ic\_menu***);  
 ab.setDisplayHomeAsUpEnabled(**true**);  
   
   
 *// Set up the navigation drawer.* **mDrawerLayout** = (DrawerLayout) findViewById(R.id.***drawer\_layout***);  
 **mDrawerLayout**.setStatusBarBackground(R.color.***colorPrimaryDark***);  
 NavigationView navigationView = (NavigationView) findViewById(R.id.***nav\_view***);

**if** (navigationView != **null**) {  
 setupDrawerContent(navigationView);  
 }

}  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
 **case** android.R.id.home:  
 *// Open the navigation drawer when the home icon is selected from the toolbar.* mDrawerLayout.openDrawer(GravityCompat.START);  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
   
  
 **private void** setupDrawerContent(NavigationView navigationView) {  
 navigationView.setNavigationItemSelectedListener(  
 **new** NavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 **public boolean** onNavigationItemSelected(MenuItem menuItem) {  
 **switch** (menuItem.getItemId()) {  
 **case** R.id.***list\_navigation\_menu\_item***:  
 *// Do nothing, we're already on that screen* **break**;  
 **case** R.id.***statistics\_navigation\_menu\_item***:  
 Intent intent =  
 **new** Intent(TasksActivity.**this**, **null**);  
 startActivity(intent);  
 **break**;  
 **default**:  
 **break**;  
 }  
 *// Close the navigation drawer when an item is selected.* menuItem.setChecked(**true**);  
 **mDrawerLayout**.closeDrawers();  
 **return true**;  
 }  
 });  
 }  
}

# Splash

**protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 Observable.*just*(initialdata())  
 .subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(result -> MainActivity());  
  
}  
  
**private void** MainActivity(){  
 Intent intent = **new** Intent(**this**, MainActivity.**class**);  
 startActivity(intent);  
 finish();  
}  
  
**public** String initialdata(){  
 **return ""**;  
}

<**style name="LaunchTheme" parent="AppTheme"**>  
 <**item name="android:windowBackground"**>@drawable/splash</**item**>  
</**style**>

# Layout

## DrawerLayout

<**android.support.v4.widget.DrawerLayout  
 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:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context="com.wjk32.jmusic.MusicPlayerActivity"**>  
   
</**android.support.v4.widget.DrawerLayout**>

Ce配合navigation使用

<**android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 app:headerLayout="@layout/nav\_header"  
 app:menu="@menu/drawer"**/>

# View类

## ViewGroup

### RecyclerView

继承关系：

public class RecyclerView   
extends [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html) implements [ScrollingView](https://developer.android.com/reference/android/support/v4/view/ScrollingView.html), [NestedScrollingChild2](https://developer.android.com/reference/android/support/v4/view/NestedScrollingChild2.html)

使用方法：

1. implementation 'com.android.support:recyclerview-v7:27.1.0'

2.

<android.support.v7.widget.RecyclerView  
    android:id="@+id/recyclerview"  
    android:scrollbars="vertical"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"/>

3. Global Variable

private RecyclerView mRecyclerView;  
    private RecyclerView.Adapter mAdapter;  
    private RecyclerView.LayoutManager mLayoutManager;

// use this setting to improve performance if you know that changes  
        // in content do not change the layout size of the RecyclerView  
        mRecyclerView.setHasFixedSize(true);  
  
        // use a linear layout manager  
        mLayoutManager = new LinearLayoutManager(this);  
        mRecyclerView.setLayoutManager(mLayoutManager);  
  
        // specify an adapter (see also next example)  
        mAdapter = new MyAdapter(myDataset);  
        mRecyclerView.setAdapter(mAdapter);

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {  
    private String[] mDataset;  
  
    // Provide a reference to the views for each data item  
    // Complex data items may need more than one view per item, and  
    // you provide access to all the views for a data item in a view holder  
    public static class ViewHolder extends RecyclerView.ViewHolder {  
        // each data item is just a string in this case  
        public TextView mTextView;  
        public ViewHolder(TextView v) {  
            super(v);  
            mTextView = v;  
        }  
    }  
  
    // Provide a suitable constructor (depends on the kind of dataset)  
    public MyAdapter(String[] myDataset) {  
        mDataset = myDataset;  
    }  
  
    // Create new views (invoked by the layout manager)  
    @Override  
    public MyAdapter.ViewHolder onCreateViewHolder(ViewGroup parent,  
                                                   int viewType) {  
        // create a new view  
        TextView v = (TextView) LayoutInflater.from(parent.getContext())  
                .inflate(R.layout.my\_text\_view, parent, false);  
        ...  
        ViewHolder vh = new ViewHolder(v);  
        return vh;  
    }  
  
    // Replace the contents of a view (invoked by the layout manager)  
    @Override  
    public void onBindViewHolder(ViewHolder holder, int position) {  
        // - get element from your dataset at this position  
        // - replace the contents of the view with that element  
        holder.mTextView.setText(mDataset[position]);  
  
    }  
  
    // Return the size of your dataset (invoked by the layout manager)  
    @Override  
    public int getItemCount() {  
        return mDataset.length;  
    }  
}

#### 样例

RecyclerView探索之通过ItemDecoration实现StickyHeader效果: [链接](https://blog.csdn.net/briblue/article/details/70211942)

# PullToRefreshLayout

<https://guides.codepath.com/android/implementing-pull-to-refresh-guide>

# 快速构建MVP架构

1. **创建BaseView interface**

**public interface** BaseView<T> {  
  
 **void** setPresenter(T presenter);  
  
}

1. **创建BasePresenter interface**

**public interface** BasePresenter {  
  
 **void** start();  
  
}

1. **在module里创建Contract interface**

**public interface** TasksContract {  
  
 **interface** View **extends** BaseView<Presenter> {  
   
 }  
  
 **interface** Presenter **extends** BasePresenter {  
   
 }  
   
}

1. **Fragment 实现Contract.View ,并且持有presenter的引用**

**public class** TasksFragment **extends** Fragment **implements** TasksContract.View {  
   
 **private** TasksContract.Presenter **mPresenter**;  
   
 @Override  
 **public void** setPresenter(TasksContract.Presenter presenter) {  
 **mPresenter**=presenter;  
 }  
   
}

1. **Presenter实现Contract.Presenter并且持有View的引用**

# RecyclerView

## 基础

**package** com.wjk32.mytodoapp\_mvp.tasks;  
  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.CheckBox;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
  
**import** com.wjk32.mytodoapp\_mvp.R;  
**import** com.wjk32.mytodoapp\_mvp.data.Task;  
  
**import** java.util.List;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.recyclerview.widget.RecyclerView;  
  
*/\*\*  
 \* Created by Jikang Wang on 3/14/19.  
 \*/***public class** TasksAdapter **extends** RecyclerView.Adapter<TasksAdapter.ViewHolder> {  
  
 **private** List<Task> **mTasks**;  
 **private** OnItemClickListener **mItemClickListener**;  
   
 **public** TasksAdapter(List<Task> tasks,OnItemClickListener itemClickListener){  
 **this**.**mTasks**=tasks;  
 **this**.**mItemClickListener**=itemClickListener;  
 }  
   
 @NonNull  
 @Override  
 **public** ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  
 LinearLayout v = (LinearLayout) LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.***task\_item***, parent, **false**);  
 **final** TasksAdapter.ViewHolder vh = **new** ViewHolder(v);  
 **return** vh;  
 }  
  
 @Override  
 **public void** onBindViewHolder(@NonNull ViewHolder holder, **int** position) {  
   
 Task currentTask=**mTasks**.get(position);  
   
 TextView textView=holder.**taskView**;  
 textView.setText(currentTask.getTitleForList());  
  
 CheckBox completeCB = holder.**checkBox**;  
 completeCB.setChecked(currentTask.isCompleted());  
   
 **if**(**mItemClickListener**!=**null**){  
 holder.**taskView**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **mItemClickListener**.onTaskClick(currentTask);  
 }  
 });  
 }  
  
 completeCB.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!currentTask.isCompleted()) {  
 **mItemClickListener**.onCompleteTaskClick(currentTask);  
 } **else** {  
 **mItemClickListener**.onActivateTaskClick(currentTask);  
 }  
 }  
 });  
  
 }  
  
 @Override  
 **public int** getItemCount() {  
 **return mTasks**.size();  
 }  
  
 **public class** ViewHolder **extends** RecyclerView.ViewHolder{  
 **public** TaskView **taskView**;  
 **public** CheckBox **checkBox**;  
 **public** ViewHolder(View itemView) {  
 **super**(itemView);  
 **taskView**=(TaskView)itemView.findViewById(R.id.***task\_view***);  
 **checkBox**=(CheckBox)itemView.findViewById(R.id.***complete***);  
 }  
 }  
  
 **public interface** OnItemClickListener{  
 **void** onTaskClick(Task task);  
 **void** onCompleteTaskClick(Task task);  
 **void** onActivateTaskClick(Task task);  
 }  
  
 **public void** setOnItemClickListener(OnItemClickListener onItemClickListener) {  
 **this**.**mItemClickListener** = onItemClickListener;  
 }  
  
 **public void** setmTasks(List<Task> mTasks) {  
 **this**.**mTasks** = mTasks;  
 }  
}

## ItemTouchHelper 左滑右滑

[玩转仿探探卡片式滑动效果](https://juejin.im/entry/58c6625c570c3500583d4d7d)