# **CoordinatorLayout**

dependencies {  
    implementation 'com.android.support:design:28.0.0'  
 }

**Scrolling Activity Behavior**

@CoordinatorLayout.DefaultBehavior(

FloatingActionButton.Behavior.class)

public class FloatingActionButton extends ImageButton {

...

public static class Behavior

extends CoordinatorLayout.Behavior<FloatingActionButton> {

private boolean updateFabVisibility(

CoordinatorLayout parent, AppBarLayout appBarLayout,

FloatingActionButton child {

if (a long condition) {

// If the anchor's bottom is below the seam,

// we'll animate our FAB out

child.hide();

} else {

// Else, we'll animate our FAB back in

child.show();

}

}

}

...

}

**SwipeDismissBehavior**

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_swipe\_behavior);

mCardView = (CardView) findViewById(R.id.swype\_card);

final SwipeDismissBehavior<CardView> swipe

= new SwipeDismissBehavior();

swipe.setSwipeDirection(

SwipeDismissBehavior.SWIPE\_DIRECTION\_ANY);

swipe.setListener(

new SwipeDismissBehavior.OnDismissListener() {

@Override public void onDismiss(View view) {

Toast.makeText(SwipeBehaviorExampleActivity.this,

"Card swiped !!", Toast.LENGTH\_SHORT).show();

}

@Override

public void onDragStateChanged(int state) {}

});

LayoutParams coordinatorParams =

(LayoutParams) mCardView.getLayoutParams();

coordinatorParams.setBehavior(swipe);

}

# **RecyclerView**

dependencies {  
    implementation 'com.android.support:recyclerview-v7:28.0.0'  
 }

public class **MyAdapter** extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {  
    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 **MyViewHolder** extends RecyclerView.ViewHolder {  
        // each data item is just a string in this case  
        public TextView mTextView;  
        public MyViewHolder(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.MyViewHolder onCreateViewHolder(ViewGroup parent,  
                                                   int viewType) {  
        // create a new view  
        TextView v = (TextView) LayoutInflater.from(parent.getContext())  
                .inflate(R.layout.my\_text\_view, parent, false);  
        ...  
        MyViewHolder vh = new MyViewHolder(v);  
        return vh;  
    }

    // Replace the contents of a view (invoked by the layout manager)  
    @Override  
    public void onBindViewHolder(MyViewHolder 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;  
    }  
}

/////////////////////////////////////////////////////////////////////////////////////////////////////

public class **MyActivity** extends Activity {  
    private RecyclerView mRecyclerView;  
    private RecyclerView.Adapter mAdapter;  
    private RecyclerView.LayoutManager mLayoutManager;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.my\_activity);  
        mRecyclerView = (RecyclerView) findViewById(R.id.my\_recycler\_view);  
  
        // 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);  
    }  
    // ...  
}

# **Snackbar**

dependencies {  
    implementation 'com.android.support:design:28.0.0'  
 }

Snackbar snackbar = Snackbar.make(coordinatorLayout, "Message is deleted", Snackbar.LENGTH\_LONG)

.set**Action**("UNDO", new View.OnClickListener() {

@Override

public void onClick(View view) {

//perform an action

}

});

snackbar.show();

# **ViewPager**

dependencies {  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
 }

## **Implementing The Activity**

public class PageViewActivity extends FragmentActivity {

MyPageAdapter pageAdapter;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_page\_view);

List<Fragment> fragments = getFragments();

pageAdapter = new MyPageAdapter(getSupportFragmentManager(), fragments);

ViewPager pager =

(ViewPager)findViewById(R.id.viewpager);

pager.setAdapter(pageAdapter);

}

}

**Implementing The PageAdapter**

class MyPageAdapter extends FragmentPagerAdapter {

private List<Fragment> fragments;

public MyPageAdapter(FragmentManager fm, List<Fragment> fragments) {

super(fm);

this.fragments = fragments;

}

@Override

public Fragment getItem(int position) {

return this.fragments.get(position);

}

@Override

public int getCount() {

return this.fragments.size();

}

}

**Getting The Fragments Set Up**

private List<Fragment> getFragments(){

List<Fragment> fList = new ArrayList<Fragment>();

fList.add(MyFragment.newInstance("Fragment 1"));

fList.add(MyFragment.newInstance("Fragment 2"));

fList.add(MyFragment.newInstance("Fragment 3"));

return fList;

//////////////////////////////////////////////////////////////////////////////////////////

public class MyFragment extends Fragment {

public static final String EXTRA\_MESSAGE = "EXTRA\_MESSAGE";

public static final MyFragment newInstance(String message)

{

MyFragment f = new MyFragment();

Bundle bdl = new Bundle(1);

bdl.putString(EXTRA\_MESSAGE, message);

f.setArguments(bdl);

return f;

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

String message = getArguments().getString(EXTRA\_MESSAGE);

View v = inflater.inflate(R.layout.myfragment\_layout, container, false);

TextView messageTextView = (TextView)v.findViewById(R.id.textView);

messageTextView.setText(message);

return v;

}

}