
● CoordinatorLayout: 뷰 간의 상호 작용을 목적으로 만든 레이아웃이다. 대표적인 사용 예는 화면에 함께 보이는 여러 뷰 가운데 하나의 뷰가 드레그될 때 다른 뷰가 함께 드레그되는 경우이다. 안드로이드에서 제공하는 모든 뷰의 상호 연동을 지원하는 것은 아니고, 주로 NestedScrollView와 AppBar의 상호 연동, AppBar와 RecyclerView의 상호 연동 부분에 이용된다.

▶ CoordinatorLayout 이용해서 AppBar와 RecyclerView 연동하기: 앱이 AppBar을 제공하면 툴바를 포함하여 화면 상단을 더 다양하게 꾸밀 수 있지만, 그만큼 RecyclerView의 데이터가 보이는 부분이 줄어들게 된다. 그러므로 사용자가 RecyclerView를 위아래로 스크롤 할 때 AppBar도 함께 스크롤되도록 하여, 사용자가 RecyclerView 내용을 조금 더 넓게 볼 수 있게 해주어야 하는데, 이를 CoordinatorLayout에서 지원한다. 구현 시에 일단 전체를 CoordinatorLayout으로 감싸야 하며, 그 하위에 AppBarLayout과 RecyclerView를 포함해야 한다.

```
// activity_main.xml

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="192dp"
        android:id="@+id/appbar">

        <com.google.android.material.appbar.CollapsingToolbarLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed">

            <androidx.constraintlayout.utils.widget.ImageFilterView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:layout_collapseMode="parallax"/>

            <androidx.appcompat.widget.Toolbar
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:layout_collapseMode="pin"/>

        </com.google.android.material.appbar.CollapsingToolbarLayout>

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

사용자의 하면 스크롤은 RecyclerView에서 발생하므로, 이 RecyclerView의 스크린 정보가 AppBarLayout에 전달되어야 AppBar가 함께 스크롤된다. 이 역할은 RecyclerView의 layout_behaviour 속성에서 설정한 behaviour에 의해 이루어진다. 문자열 리소스로 값을 지정하였지만, 일반 문자열이 아니라 다음과 같은 클래스명이다.

```
com.google.android.material.appbar.AppBarLayout$ScrollingViewBehaviour
```

결국, 스크롤 정보가 ScrollingViewBehaviour에 전달되어 ScrollingViewBehaviour에 의해 AppBarLayout이 조정되기 때문에, RecyclerView와 AppBarLayout이 함께 스크롤된다. ScrollingViewBehaviour은 Behavior 클래스의 일종으로 이해하면 되며, Behavior 클래스는 특정 뷰에 의존하여 다른 어떤 행동에 따라 뷰를 변경하는 것을 주목적으로 하는 클래스이다.

AppBarLayout이 스크롤 될 때, CollapsingToolBarLayout을 이용하여 AppBar가 어떻게 스크롤 되어야 하는지 설정할 수 있다. CollapsingToolBarLayout의 layout_scrollFlags 속성으로 AppBar의 스크롤 여부와 방식을 지정한다.

CollapsingToolBarLayout layout_scrollFlags="여기 들어갈 속성 값"	AppBar의 스크롤 여부와 방식을 지정
scroll	스크롤 되게 지정. RecyclerView가 스크롤될 때 AppBarLayout 부분이 함께 스크롤되게 한다.
exitUntilCollapsed	minHeight(ToolBar의 세로 크기)까지만 스크롤 되게 지정
enterAlwaysCollapsed	스크롤이되어 전체가 사라지게 지정
scroll exitUntilCollapsed	스크롤되다 ToolBar 정도의 크기만 남으면 더는 스크롤되지 않음
scroll enterAlwaysCollapsed	전체가 스크롤되어 사라짐

▶ CollapsingToolBarLayout의 contentScrim 속성: AppBarLayout의 구성에 ImageView가 있으면 스크롤 때문에 이미지가 사라지기 시작한다. 계속 스크롤되어 크기가 줄면서 이미지가 정상적으로 나오지 않을 때, 이미지의 일부분만 보이게 하지 않고 전체 AppBar을 contentScrim 속성에서 지정한 색으로 채울 수 있다.

```
pp:contentScrim="?attr/colorPrimary"
```

▶ CollapsingToolBarLayout 하위에 있는 ToolBar와 ImageView의 layout_collapseMode 속성: 각 뷰가 스크롤 될 때 어떻게 움직여야 하는지를 지정하는 속성이다.

CollapsingToolBarLayout 하위의 ToolBar과 ImageView layout_collapseMode	각 뷰가 스크롤될 때 어떻게 움직여야 하는지 지정하는 속성
parallax	초기 스크롤부터 함께 스크롤되게 지정
pin	초기 고정 상태로 스크롤 되지 않게 지정

```
// activity_main.xml

<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="192dp"
        android:id="@+id/appbar">

        <com.google.android.material.appbar.CollapsingToolbarLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/collapsing"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed"
            app:expandedTitleMarginBottom="48dp"
            app:expandedTitleMarginStart="48dp"
            app:title="Hello kkang!!">

            <androidx.constraintlayout.utils.widget.ImageFilterView
                android:id="@+id/app_bar_image"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:layout_collapseMode="parallax"
                android:src="@drawable/appbar_image"
                android:scaleType="centerCrop"/>

            <androidx.appcompat.widget.Toolbar
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                app:layout_collapseMode="pin"
                android:id="@+id/toolbar"/>

        </com.google.android.material.appbar.CollapsingToolbarLayout>

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        android:id="@+id/recyclerView"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

```
// MainActivity.java

package com.android.practicecoordinatorlayout;

import android.os.Bundle;
import android.view.LayoutInflater;
```

```

import android.view.Menu;
import android.view.ViewGroup;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.android.practicecoordinatorlayout.databinding.ActivityMainBinding;
import com.android.practicecoordinatorlayout.databinding.ItemBinding;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ActivityMainBinding binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        List<String> list = new ArrayList<>();
        for (int i = 0; i < 20; i++) {
            list.add("Item=" + i);
        }
        binding.recyclerView.setLayoutManager(new LinearLayoutManager(this));
        binding.recyclerView.setAdapter(new MyAdapter(list));
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_appbar, menu);
        return super.onCreateOptionsMenu(menu);
    }

    private class MyAdapter extends RecyclerView.Adapter<MyViewHolder> {
        private List<String> list;

        public MyAdapter(List<String> list) {
            this.list = list;
        }

        @Override
        public MyViewHolder onCreateViewHolder(ViewGroup viewGroup, int i) {
            ItemBinding binding = ItemBinding.inflate(LayoutInflater.from(viewGroup.getContext()));
            return new MyViewHolder(binding);
        }

        @Override
        public void onBindViewHolder(MyViewHolder viewHolder, int position) {
            String text = list.get(position);
            viewHolder.binding.itemTextView.setText(text);
        }

        @Override
        public int getItemCount() {

```

```
        return list.size();
    }
}

private class MyViewHolder extends RecyclerView.ViewHolder {
    ItemBinding binding;

    public MyViewHolder(ItemBinding binding) {
        super(binding.getRoot());
        this.binding = binding;
    }
}
}
```

Resources: 강샘의 안드로이드 프로그래밍
