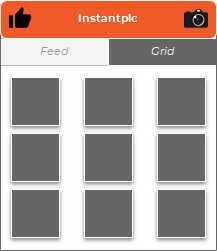
**Lab 9: Android GridView Layout**

# **Introduction**

Items are shown in a two-dimensional scrolling grid via Android GridView (with rows & columns). The grid items could be dynamically inserted using a ListAdapter rather than being preset**.**

****

|  |  |
| --- | --- |
| Attribute | Description |
| id | Used to identify uniquely |
| columnWidth | Used to specify width of each columns. It can be specified in sp, in, or mm, px, dp, |
| gravity | Specify the gravity in each cell. |
| horizontalSpacing | Used to specify spacing between cells horizontally |
| numColumns | Specify the number of columns to show. |

**Let’s get Started**

This exercise will take you through simple steps to show how to create your own Android application using GridView Layout.

**Step 1: Create a New Project in Android Studio as shown below**

Graphical user interface, text, application

Description automatically generated

**Step 2: Select Empty Activity as shown below**

Graphical user interface, application, shape

Description automatically generated

**Step 3: Update MainActivity.java as per the code given below**

**package** com.example.demogridlayout;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.GridView;  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 GridView gridview = (GridView) findViewById(R.id.***gridview***);  
 gridview.setAdapter(**new** ImageAdapter(**this**));  
 gridview.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 **public void** onItemClick(AdapterView<?> parent,  
 View v, **int** position, **long** id){  
 *// Send intent to SingleViewActivity* Intent i = **new** Intent(getApplicationContext(), SingleViewActivity.**class**);  
 *// Pass image index* i.putExtra(**"id"**, position);  
 startActivity(i);  
 }  
 });  
 }  
}

**Step 4: Update activity\_main.xml as per the code given below**

*<?***xml version="1.0" encoding="utf-8"***?>*<**GridView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/gridview"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:columnWidth="90dp"  
 android:numColumns="3"  
 android:verticalSpacing="10dp"  
 android:horizontalSpacing="10dp"  
 android:stretchMode="columnWidth"  
 android:gravity="center"** />

**Step 5: Create ImageAdapter.java as per the code given below**

**package** com.example.demogridlayout;  
**import** android.content.Context;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.BaseAdapter;  
**import** android.widget.GridView;  
**import** android.widget.ImageView;  
  
**public class** ImageAdapter **extends** BaseAdapter {  
 **private** Context **mContext**;  
  
 *// Constructor* **public** ImageAdapter(Context c) {  
 **mContext** = c;  
 }  
  
 **public int** getCount() {  
 **return mThumbIds**.**length**;  
 }  
  
 **public** Object getItem(**int** position) {  
 **return null**;  
 }  
  
 **public long** getItemId(**int** position) {  
 **return** 0;  
 }  
  
 *// create a new ImageView for each item referenced by the Adapter* **public** View getView(**int** position, View convertView, ViewGroup parent) {  
 ImageView imageView;  
  
 **if** (convertView == **null**) {  
 imageView = **new** ImageView(**mContext**);  
 imageView.setLayoutParams(**new** GridView.LayoutParams(185, 185));  
 imageView.setScaleType(ImageView.ScaleType.***CENTER\_CROP***);  
 imageView.setPadding(8, 8, 8, 8);  
 }  
 **else** {  
 imageView = (ImageView) convertView;  
 }  
 imageView.setImageResource(**mThumbIds**[position]);  
 **return** imageView;  
 }  
  
 *// Keep all Images in array* **public** Integer[] **mThumbIds** = {  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 R.drawable.***myimg1***, R.drawable.***myimg2***,  
 };  
}

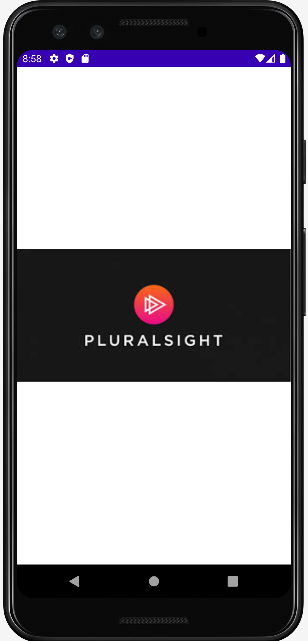
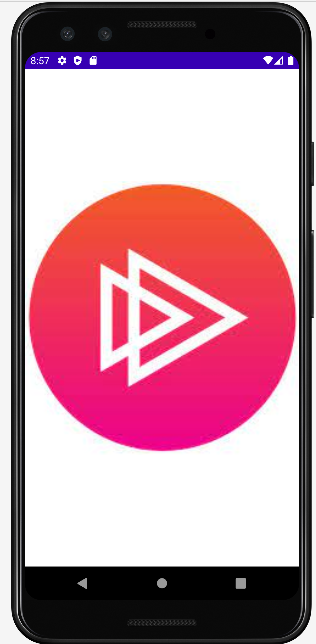
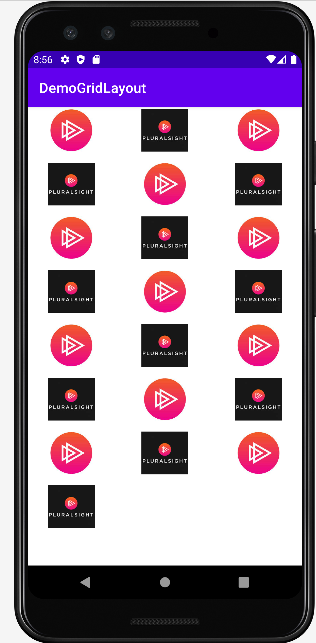
**Step 6: Create SingleViewActivity.java as per the code given below**

**package** com.example.demogridlayout;  
**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.widget.ImageView;  
  
**public class** SingleViewActivity **extends** Activity {  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***single\_view***);  
  
 *// Get intent data* Intent i = getIntent();  
  
 *// Selected image id* **int** position = i.getExtras().getInt(**"id"**);  
 ImageAdapter imageAdapter = **new** ImageAdapter(**this**);  
  
 ImageView imageView = (ImageView) findViewById(R.id.***SingleView***);  
 imageView.setImageResource(imageAdapter.**mThumbIds**[position]);  
 }  
}

**Step 7: Create single\_view.xml Layout file as per the code given below**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"** >  
  
 <**ImageView android:id="@+id/SingleView"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"**/>  
  
</**LinearLayout**>

**Step 8: Check Output on Android Emulator.**



**Voila!!** We have successfully completed this lab.