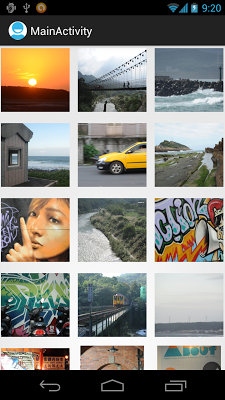
**[Android-er](http://android-er.blogspot.in/)**

**GridView loading photos from SD Card**

Previous exercises described how to implement "[Gallery-like HorizontalScrollView](http://android-er.blogspot.com/2012/07/implement-custom-linearlayout-for.html)" and "[Vertical Gallery-like ScrollView](http://android-er.blogspot.com/2012/07/vertical-gallery-like-scrollview.html)". Alternatively, it can be displayed in GridView.

[](http://2.bp.blogspot.com/-VEoY5WMozUI/UBU5eHp8unI/AAAAAAAAD7g/JWDAqboJbeo/s1600/GridView_01.png)

Add a <GridView> in layout.

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<GridView

android:id="@+id/gridview"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:columnWidth="90dp"

android:numColumns="auto\_fit"

android:verticalSpacing="10dp"

android:horizontalSpacing="10dp"

android:stretchMode="columnWidth"

android:gravity="center"/>

</LinearLayout>

Main code:

package com.example.androidgridview;

import java.io.File;

import java.util.ArrayList;

import android.os.Bundle;

import android.os.Environment;

import android.app.Activity;

import android.content.Context;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.GridView;

import android.widget.ImageView;

import android.widget.Toast;

public class MainActivity extends Activity {

public class ImageAdapter extends BaseAdapter {

private Context mContext;

ArrayList<String> itemList = new ArrayList<String>();

public ImageAdapter(Context c) {

mContext = c;

}

void add(String path){

itemList.add(path);

}

@Override

public int getCount() {

return itemList.size();

}

@Override

public Object getItem(int arg0) {

// TODO Auto-generated method stub

return null;

}

@Override

public long getItemId(int position) {

// TODO Auto-generated method stub

return 0;

}

@Override

public View getView(int position, View convertView, ViewGroup parent) {

ImageView imageView;

if (convertView == null) { // if it's not recycled, initialize some attributes

imageView = new ImageView(mContext);

imageView.setLayoutParams(new GridView.LayoutParams(220, 220));

imageView.setScaleType(ImageView.ScaleType.CENTER\_CROP);

imageView.setPadding(8, 8, 8, 8);

} else {

imageView = (ImageView) convertView;

}

Bitmap bm = decodeSampledBitmapFromUri(itemList.get(position), 220, 220);

imageView.setImageBitmap(bm);

return imageView;

}

public **Bitmap decodeSampledBitmapFromUri(String path, int reqWidth, int reqHeight) {**

**Bitmap bm = null;**

**// First decode with inJustDecodeBounds=true to check dimensions**

**final BitmapFactory.Options options = new BitmapFactory.Options();**

**options.inJustDecodeBounds = true;**

**BitmapFactory.decodeFile(path, options);**

**// Calculate inSampleSize**

**options.inSampleSize = calculateInSampleSize(options, reqWidth, reqHeight);**

**// Decode bitmap with inSampleSize set**

**options.inJustDecodeBounds = false;**

**bm = BitmapFactory.decodeFile(path, options);**

**return bm;**

**}**

public int calculateInSampleSize(

BitmapFactory.Options options, int reqWidth, int reqHeight) {

// Raw height and width of image

final int height = options.outHeight;

final int width = options.outWidth;

int inSampleSize = 1;

if (height > reqHeight || width > reqWidth) {

if (width > height) {

inSampleSize = Math.round((float)height / (float)reqHeight);

} else {

inSampleSize = Math.round((float)width / (float)reqWidth);

}

}

return inSampleSize;

}

}

ImageAdapter myImageAdapter;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

GridView gridview = (GridView) findViewById(R.id.gridview);

myImageAdapter = new ImageAdapter(this);

gridview.setAdapter(myImageAdapter);

String ExternalStorageDirectoryPath = Environment

.getExternalStorageDirectory()

.getAbsolutePath();

String targetPath = ExternalStorageDirectoryPath + "/test/";

Toast.makeText(getApplicationContext(), targetPath, Toast.LENGTH\_LONG).show();

File targetDirector = new File(targetPath);

File[] files = targetDirector.listFiles();

for (File file : files){

myImageAdapter.add(file.getAbsolutePath());

}

}

}