**Displaying images from SD card in Android**

03.31.2011

| 80025 views |

Below you will find a Android example of how to access and display images that are stored on your SD card.  
  
I wrote [part 2](http://mihaifonoage.blogspot.com/2009/11/displaying-images-from-sd-card-in.html) for this article, where images are loaded in the background using an asynchronous task. It is an improvement over this article, but I strongly suggest trying this one first to fully appreciate the differences between the two approaches.  
  
The main idea is to make use of the MediaStore class, which is a Media provider that contains data for all available media on both internal and external storage devices (such as an SD card). An adapter is used as a bridge between the data and the view.  
  
The activity is shown below:

[view source](http://mobile.dzone.com/news/displaying-images-sd-card" \l "viewSource" \o "view source)

[print](http://mobile.dzone.com/news/displaying-images-sd-card" \l "printSource" \o "print)[?](http://mobile.dzone.com/news/displaying-images-sd-card" \l "about" \o "?)

001.package blog.android.sdcard;

002.

003.import android.app.Activity;

004.import android.content.Context;

005.import android.content.Intent;

006.import android.database.Cursor;

007.import android.net.Uri;

008.import android.os.Bundle;

009.import android.provider.MediaStore;

010.import android.view.Menu;

011.import android.view.MenuInflater;

012.import android.view.MenuItem;

013.import android.view.View;

014.import android.view.ViewGroup;

015.import android.widget.AdapterView;

016.import android.widget.BaseAdapter;

017.import android.widget.GridView;

018.import android.widget.ImageView;

019.import android.widget.AdapterView.OnItemClickListener;

020.

021./\*\*

022.\* Displays images from an SD card.

023.\*/

024.public class SDCardImagesActivity extends Activity {

025.

026./\*\*

027.\* Cursor used to access the results from querying for images on the SD card.

028.\*/

029.private Cursor cursor;

030./\*

031.\* Column index for the Thumbnails Image IDs.

032.\*/

033.private int columnIndex;

034.

035.@Override

036.public void onCreate(Bundle savedInstanceState) {

037.super.onCreate(savedInstanceState);

038.setContentView(R.layout.sdcard);

039.

040.// Set up an array of the Thumbnail Image ID column we want

041.String[] projection = {MediaStore.Images.Thumbnails.\_ID};

042.// Create the cursor pointing to the SDCard

043.cursor = managedQuery( MediaStore.Images.Thumbnails.EXTERNAL\_CONTENT\_URI,

044.projection, // Which columns to return

045.null,       // Return all rows

046.null,

047.MediaStore.Images.Thumbnails.IMAGE\_ID);

048.// Get the column index of the Thumbnails Image ID

049.columnIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Thumbnails.\_ID);

050.

051.GridView sdcardImages = (GridView) findViewById(R.id.sdcard);

052.sdcardImages.setAdapter(new ImageAdapter(this));

053.

054.// Set up a click listener

055.sdcardImages.setOnItemClickListener(new OnItemClickListener() {

056.public void onItemClick(AdapterView parent, View v, int position, long id) {

057.// Get the data location of the image

058.String[] projection = {MediaStore.Images.Media.DATA};

059.cursor = managedQuery( MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI,

060.projection, // Which columns to return

061.null,       // Return all rows

062.null,

063.null);

064.columnIndex = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);

065.cursor.moveToPosition(position);

066.// Get image filename

067.String imagePath = cursor.getString(columnIndex);

068.// Use this path to do further processing, i.e. full screen display

069.}

070.});

071.}

072.

073./\*\*

074.\* Adapter for our image files.

075.\*/

076.private class ImageAdapter extends BaseAdapter {

077.

078.private Context context;

079.

080.public ImageAdapter(Context localContext) {

081.context = localContext;

082.}

083.

084.public int getCount() {

085.return cursor.getCount();

086.}

087.public Object getItem(int position) {

088.return position;

089.}

090.public long getItemId(int position) {

091.return position;

092.}

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

094.ImageView picturesView;

095.if (convertView == null) {

096.picturesView = new ImageView(context);

097.// Move cursor to current position

098.cursor.moveToPosition(position);

099.// Get the current value for the requested column

100.int imageID = cursor.getInt(columnIndex);

101.// Set the content of the image based on the provided URI

102.picturesView.setImageURI(Uri.withAppendedPath(

103.MediaStore.Images.Thumbnails.EXTERNAL\_CONTENT\_URI, "" + imageID));

104.picturesView.setScaleType(ImageView.ScaleType.FIT\_CENTER);

105.picturesView.setPadding(8, 8, 8, 8);

106.picturesView.setLayoutParams(new GridView.LayoutParams(100, 100));

107.}

108.else {

109.picturesView = (ImageView)convertView;

110.}

111.return picturesView;

112.}

113.}

114.}

The layout of the main activity is shown below:

[view source](http://mobile.dzone.com/news/displaying-images-sd-card" \l "viewSource" \o "view source)

[print](http://mobile.dzone.com/news/displaying-images-sd-card" \l "printSource" \o "print)[?](http://mobile.dzone.com/news/displaying-images-sd-card" \l "about" \o "?)

01.<?xml version="1.0" encoding="utf-8"?>

02.<GridView xmlns:android="<http://schemas.android.com/apk/res/android>"

03.android:id="@+id/sdcard"

04.android:layout\_width="fill\_parent"

05.android:layout\_height="fill\_parent"

06.android:padding="10dp"

07.android:verticalSpacing="10dp"

08.android:horizontalSpacing="10dp"

09.android:numColumns="auto\_fit"

10.android:columnWidth="90dp"

11.android:stretchMode="columnWidth"

12.android:gravity="center"

13./>

In order for this to work, you need to [emulate an SD card](http://www.anddev.org/emulating_a_sd-card-t263.html).  
  
Enjoy!  
  
UPDATE: In order to be bale to view thumbnails images from the SD Card, Android needs to create them first, hence you should start the Gallery application that comes preinstalled, and open the sdcard folder which will automatically create thumbnails for the images stored on your sdcard. This is a current shortcoming of the SDK that will be fixed in future releases (<http://groups.google.com/group/android-developers/browse_thread/thread/3f01b284e2537312/fa9487d19db4907e>).  
  
UPDATE: For some reason, if you use

MediaStore.Images.Thumbnails.IMAGE\_ID

like in the previous version of the above code, the images are not *always* displayed on the screen. Changing to

MediaStore.Images.Thumbnails.\_ID

seems to solve the problem. I will look more into why and get back to you.  
Furthermore, some images have the wrong path attached to them.  I changed the creation of the cursor object from

[view source](http://mobile.dzone.com/news/displaying-images-sd-card" \l "viewSource" \o "view source)

[print](http://mobile.dzone.com/news/displaying-images-sd-card" \l "printSource" \o "print)[?](http://mobile.dzone.com/news/displaying-images-sd-card" \l "about" \o "?)

1.cursor = managedQuery( MediaStore.Images.Thumbnails.EXTERNAL\_CONTENT\_URI,

2.projection, // Which columns to return

3.null,       // Return all rows

4.null,

5.null);

to

[view source](http://mobile.dzone.com/news/displaying-images-sd-card" \l "viewSource" \o "view source)

[print](http://mobile.dzone.com/news/displaying-images-sd-card" \l "printSource" \o "print)[?](http://mobile.dzone.com/news/displaying-images-sd-card" \l "about" \o "?)

1.cursor = managedQuery( MediaStore.Images.Thumbnails.EXTERNAL\_CONTENT\_URI,

2.projection, // Which columns to return

3.null,       // Return all rows

4.null,

5.MediaStore.Images.Thumbnails.IMAGE\_ID);

References

Reference:

[Displaying images from SD card in Android](http://mihaifonoage.blogspot.com/2009/09/displaying-images-from-sd-card-in.html)

Published at DZone with permission of [Mihai Fonoage](http://mobile.dzone.com/users/fonoage81), author and DZone MVB. ([source](http://mihaifonoage.blogspot.com/2009/09/displaying-images-from-sd-card-in.html" \t "_blank))

*(Note: Opinions expressed in this article and its replies are the opinions of their respective authors and not those of DZone, Inc.)*

Tags:

* [android](http://mobile.dzone.com/category/tags/android" \o ")
* [Java](http://mobile.dzone.com/category/dzone-taxonomy/java" \o ")