

Healtheon

简单，快乐，创新，极致.....

【专注于移动与嵌入式领域】

博客园 首页 新随笔 联系 管理 订阅 XML

随笔- 61 文章- 0 评论- 1032

公告

16,287 Pageviews
Sep. 08th - Oct. 08th

Android实现获取本机中所有图片

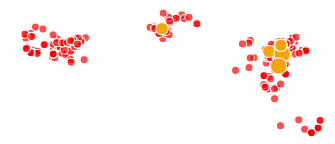
本示例演示如何在Android中使用加载器(Loader)来实现获取本机中的所有图片，并进行查看图片的效果。

在这个示例中，我使用android-support-v4.jar中的加载器(Loader)来实现获取本机中所有图片，关于这个包在以前的文章中也提到，是一个非常有用的包，关于这个包的详细信息请大家查看官方文档：<http://developer.android.com/sdk/compatibility-library.html>。

关于加载器(Loader)是在Android3.0中才开始引进并使用的，Android3.0以前的版本中要想使用加载器则需要用android-support-v4.jar来实现，我这个示例是基于Android2.2的。加载器(Loader)有什么作用呢？官方文档介绍是：它能够使用得在activity或fragment中异步加载数据变得更加容易，它具有以下的特点：

1. 它们对每一个Activity和Fragment都是有效的。
2. 它们提供了一种异步加载数据的能力。
3. 它们监视数据源并且数据内容改变时将会传递新的结果。
4. 当配置改变而被重新创建时，它们自动的会重连到上一个加载器的游标，然而，它们不需要重新查询数据。

我会结合下面的代码逐步介绍加载器(Loader)相关类及使用方法。下面先让我们看下本示例实现的效果图：



总访问量：**2244911**

每日统计：

站长统计

今日IP[]

今日PV[]

昨日IP[]

昨日PV[]

当前在线[5]

百度Hi：hspl_2008

微信账号：Healtheon

联系方式：

Email：hanyonglu@126.com

Gmail：hanyonglu@gmail.com

新浪微博：

<http://weibo.com/hanyonglu>

腾讯微博：

<http://t.qq.com/hanyonglu2010/>

公众账号：YiChengKanKan



开源项目：
<https://github.com/hanyonglu>
昵称：[Healtheon](#)
园龄：[8年6个月](#)
粉丝：[910](#)
关注：[4](#)
[+加关注](#)

<	2012年5月						>
日	一	二	三	四	五	六	
29	30	1	2	3	4	5	
6	7	8	9	10	11	12	
13	14	15	16	17	18	19	
20	21	22	23	24	25	26	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	

搜索

找找看

谷歌搜索

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)
[更多链接](#)

最新随笔

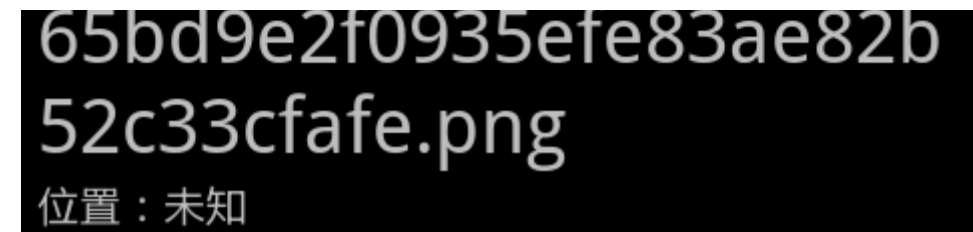
1. [Android性能优化典范第二季](#)
2. [Android性能优化典范第一季](#)
3. [Android实现伸缩弹力分布菜单效果](#)
4. [JAVA实现随机无重复数字功能](#)
5. [Android实现甘特图\(GanttChart\)效果](#)
6. [Android实现宫格图片连续滑动效果](#)
7. [Android in Mono开发初体验之DataBase](#)
8. [Android in Mono开发初体验之ListAdapter](#)
9. [Android实现下拉导航选择菜单效果](#)
10. [Android实现局部图片滑动指引效果](#)

我的标签

[Android性能\(2\)](#)
[电量优化\(2\)](#)
[缓存\(1\)](#)
[内存优化与GC\(1\)](#)
[使用对象池来提高效率\(1\)](#)
[网络优化\(1\)](#)
[自定义View性能\(1\)](#)
[Bitmap的缩放\(1\)](#)
[Google移动研发\(1\)](#)
[LRU Cache\(1\)](#)
[更多](#)

随笔分类(60)

[【Algorithm Analysis】\(1\)](#)





左边的图是显示本机中所有的图片列表，右边的图则是点击一项时查看图片。

项目结构图如下所示：

[【Android Develop】\(37\)](#)
[【Architecture & Design】](#)
[【C & C++】\(3\)](#)
[【Computer Graphics】](#)
[【Dotnet Framework】](#)
[【Dotnet Technology】\(6\)](#)
[【IOS Develop】\(1\)](#)
[【JAVA Develop】](#)
[【Linux Kernel】](#)
[【Linux Program】](#)
[【MonoDevelop】\(2\)](#)
[【Network Service】\(4\)](#)
[【OpenGL & ES】\(2\)](#)
[【Win32/64 SDK & OS】\(4\)](#)

随笔档案⁽⁶¹⁾

[2015年5月 \(1\)](#)
[2015年1月 \(1\)](#)
[2012年10月 \(2\)](#)
[2012年9月 \(2\)](#)
[2012年8月 \(2\)](#)
[2012年7月 \(1\)](#)
[2012年6月 \(2\)](#)
[2012年5月 \(2\)](#)
[2012年4月 \(6\)](#)
[2012年3月 \(13\)](#)
[2012年2月 \(12\)](#)
[2011年6月 \(1\)](#)
[2011年5月 \(1\)](#)
[2011年4月 \(15\)](#)

推荐书籍

计算机科学丛书



创建项目时我们需要引入android-support-v4.jar包。

MyDevicePhotoActivity.java文件中代码如下：

```

package com.device.photo;

import android.app.Dialog;

```

亚马逊
amazon.cn

图书
音像
软件

消费电子
日用百货

更多.....

[点击这里](#)

隐私声明

友情链接

[Trinea](#)

```

import android.content.ContentResolver;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;

import android.support.v4.app.FragmentActivity;
import android.support.v4.app.LoaderManager.LoaderCallbacks;
import android.support.v4.content.CursorLoader;
import android.support.v4.content.Loader;
import android.support.v4.widget.SimpleCursorAdapter;
import android.support.v4.widget.SimpleCursorAdapter.ViewBinder;

/**
 * Android实现获取本机中所有图片
 * @Description: Android实现获取本机中所有图片
 *
 * @FileName: MyDevicePhotoActivity.java
 *
 * @Package com.device.photo
 *
 * @Author Hanyonglu
 *
 * @Date 2012-5-10 下午04:43:55
 *
 * @Version V1.0
 */
public class MyDevicePhotoActivity extends FragmentActivity implements LoaderCallbacks<Cursor>{
    private Bitmap bitmap = null;
    private byte[] mContent = null;

```

A User, An Androider, A Runner, A Biker, A Dreamer

[极光推送博客](#)

[计算机与互联网](#)

积分与排名

积分 - 152114

排名 - 1595

最新评论

1. [Re:Android开发之APN网络切换](#)

@xhh666你好，很久都没有维护这个博客了，demo我也找不到了。麻烦您百度上再搜索一下吧。 ...

--Healtheon

2. [Re:Android开发之APN网络切换](#)

示例代码可以发我一份吗？谢啦！1185719419@qq.com

--xhh666

3. [Re:C/C++堆、栈及静态数据区详解](#)

void add(int); int num; main() { int n=5; add(n.....

--cnbgs

4. [Re:va_start和va_end使用详解](#)

@NanoDragon这是因为vsprintf 可以接收va_list ap，已经有格式化打印的实现，格式化打印参数字符串已经约定好他后面的其他参数必须是什么类型，如过类型不一致会编译不过的。实际.....

--linuxdog

5. [Re:Android实现宫格图片连续滑动效果](#)

语音记帐软件：语音实现在线听书用Kotlin开发android平台语音识别，语义理解应用 (olamisdsk) ...

--ls0609

```
private ListView listView = null;
private SimpleCursorAdapter simpleCursorAdapter = null;

private static final String[] STORE_IMAGES = {
    MediaStore.Images.Media.DISPLAY_NAME,
    MediaStore.Images.Media.LATITUDE,
    MediaStore.Images.Media.LONGITUDE,
    MediaStore.Images.Media._ID
};

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    listView = (ListView)findViewById(android.R.id.list);
    simpleCursorAdapter = new SimpleCursorAdapter(
        this,
        R.layout.simple_list_item,
        null,
        STORE_IMAGES,
        new int[] { R.id.item_title, R.id.item_value},
        0
    );

    simpleCursorAdapter.setViewBinder(new ImageLocationBinder());
    listView.setAdapter(simpleCursorAdapter);
    // 注意此处是getSupportLoaderManager(), 而不是getLoaderManager()方法。
    getSupportLoaderManager().initLoader(0, null, this);

    // 单击显示图片
    listView.setOnItemClickListener(new ShowItemImageOnClickListener());
}

@Override
public Loader<Cursor> onCreateLoader(int arg0, Bundle arg1) {
    // TODO Auto-generated method stub
```

阅读排行榜

1. [Android实现推送方式解决方案\(297514\)](#)
2. [Android操作HTTP实现与服务器通信\(136902\)](#)
3. [XMPP协议实现原理介绍\(113133\)](#)
4. [va_start和va_end使用详解\(105246\)](#)
5. [Android实现左右滑动指引效果\(92862\)](#)
6. [Android开发之日历控件实现\(92321\)](#)
7. [Android实现数据存储技术\(85471\)](#)
8. [Android开发之Intent.Action\(81311\)](#)
9. [Android实现网络多线程断点续传下载\(75050\)](#)
10. [Android实现左右滑动效果\(74390\)](#)

评论排行榜

1. [Android实现下拉导航选择菜单效果\(155\)](#)
2. [Android实现左右滑动指引效果\(111\)](#)
3. [Android Push Notification实现信息推送使用\(79\)](#)
4. [Android实现局部图片滑动指引效果\(67\)](#)
5. [Android实现导航菜单左右滑动效果\(59\)](#)
6. [Android实现程序前后台切换效果\(54\)](#)
7. [Android实现推送方式解决方案\(45\)](#)
8. [Android开发之APN网络切换\(40\)](#)
9. [JAVA实现随机无重复数字功能\(37\)](#)
10. [Android实现网络多线程断点续传下载\(36\)](#)

推荐排行榜

1. [Android实现数据存储技术\(32\)](#)
2. [Android实现推送方式解决方案\(31\)](#)
3. [Android开发之InstanceState详解\(27\)](#)
4. [Android实现左右滑动指引效果\(22\)](#)
5. [XMPP协议实现原理介绍\(21\)](#)


```
// 为了查看信息,需要用到CursorLoader。
CursorLoader cursorLoader = new CursorLoader(
    this,
    MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
    STORE_IMAGES,
    null,
    null,
    null);
return cursorLoader;
}

@Override
public void onLoaderReset(Loader<Cursor> arg0) {
    // TODO Auto-generated method stub
    simpleCursorAdapter.swapCursor(null);
}

@Override
public void onLoadFinished(Loader<Cursor> arg0, Cursor cursor) {
    // TODO Auto-generated method stub
    // 使用swapCursor()方法,以使旧的游标不被关闭。
    simpleCursorAdapter.swapCursor(cursor);
}

// 将图片的位置绑定到视图
private class ImageLocationBinder implements ViewBinder{
    @Override
    public boolean setViewValue(View view, Cursor cursor, int arg2) {
        // TODO Auto-generated method stub
        if (arg2 == 1) {
            // 图片经度和纬度
            double latitude = cursor.getDouble(arg2);
            double longitude = cursor.getDouble(arg2 + 1);

            if (latitude == 0.0 && longitude == 0.0) {
                ((TextView)view).setText("位置:未知");
            } else {
                ((TextView)view).setText("位置:" + latitude + ", " + longitude);
            }
        }
    }
}
```

- 6. Android实现网络多线程断点续传下载(21)
- 7. Android操作HTTP实现与服务器通信(19)
- 8. va_start和va_end使用详解(17)
- 9. Android实现局部图片滑动指引效果(16)
- 10. Android进程与线程基本知识(16)

// 需要注意：在使用ViewBinder绑定数据时，必须返回真；否则，SimpleCursorAdapter将会用自己的方式绑定数据。

```
        return true;
    } else {
        return false;
    }
}
```

// 单击项显示图片事件监听器

```
private class ShowItemImageOnClickListener implements OnItemClickListener{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
        long id) {
        // TODO Auto-generated method stub
        final Dialog dialog = new Dialog(MyDevicePhotoActivity.this);
        // 以对话框形式显示图片
        dialog setContentView(R.layout.image_show);
        dialog.setTitle("图片显示");

        ImageView ivImageShow = (ImageView) dialog.findViewById(R.id.ivImageShow);
        Button btnClose = (Button) dialog.findViewById(R.id.btnClose);

        btnClose.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                dialog.dismiss();

                // 释放资源
                if(bitmap != null){
                    bitmap.recycle();
                }
            }
        });

        Uri uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI.buildUpon().
            appendPath(Long.toString(id)).build();
```

```
FileUtil file = new FileUtil();
ContentResolver resolver = getContentResolver();

// 从Uri中读取图片资源
try {
    mContent =
file.readInputStream(resolver.openInputStream(Uri.parse(uri.toString())));
    bitmap = file.getBitmapFromBytes(mContent, null);
    ivImageShow.setImageBitmap(bitmap);
} catch (Exception e) {
    // TODO: handle exception
    e.printStackTrace();
}

dialog.show();
}
}

@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();

    if(bitmap != null){
        bitmap.recycle();
    }
}
}
```



我们在上面的代码中可以使用回调方法onCreateLoader()来创建一个新的加载器。

LoaderManager.LoaderCallbacks：是一个用于客户端与LoaderManager交互的回调接口。

LoaderManager：是一个抽象类，并关联到一个Activity或Fragment，管理一个或多个装载器的实例。这能够帮助一个应用管理那些与Activity或Fragment的生命周期相关的运行长时间的的操作。最常见的方式是它与一个CursorLoader一起使用，

然而应用是可以自由的写它们自己的装载器以加载其它类型的数据。

Loader：是一个执行异步数据加载的抽象类。它是加载器的基类。我们可以使用典型的CursorLoader，但是我们也可以实现自己的子类。如果加载器被激活，它们将监视它们的数据源并且当数据改变时发送新的结果。

AsyncTaskLoader：是一个抽象类，能够提供一个AsyncTask来执行异步工作。

CursorLoader：它是AsyncTaskLoader的子类，它可以查询ContentResolver然后返回一个Cursor。该类实现了加载器的协议并能够以标准的方式查询cursors.以AsyncTaskLoader为基础，在后台线程中执行cursor查询以便不阻塞程序的UI。使用这个加载器是从ContentProvider中异步查询数据的最好的方式，而不是使用Activity或Fragment的API去执行一个被管理的查询。

一个使用加载器的应用一般的会包括以下：

1. An Activity or Fragment.
2. An instance of the LoaderManager.
3. 一个被ContentProvider所支持的加载器，该加载器用以加载数据。当然了，我们也可以继承Loader或AsyncTaskLoader以实现自己的子类，去加载其他数据源的数据。
4. 实现LoaderManager.LoaderCallbacks接口，我们可以创建新的装载器以及管理已有的加载器的引用。
5. 一种展现加载器数据的方式，比如可以使用SimpleCursorAdapter。
6. 一个数据源，比如ContentProvider。

LoaderManager用以管理一个Activity或Fragment中的一个或多个装载器，但每个Activity或Fragment只能一个LoaderManager。

那么如何初始化一个加载器呢？需要在Activity的onCreate()方法或是Fragment中的onActivityCreated()方法中实现。

在本示例的代码中，通过如下代码初始化加载器。

```
getSupportLoaderManager().initLoader(0, null, this);
```

如果是在Android3.0中则需要通过如下代码进行初始化。

```
getLoaderManager().initLoader(0, null, this);
```

initLoader()方法有如下的参数：

1. 一个标识加载器的唯一ID。在本示例中，加载器的ID是0。
2. 一个可选的参数以支持加载器的构建。在本示例中，使用null。
3. 一个LoaderManager.LoaderCallbacks的实现。被LoaderManager调用以报告装载器的事件，在本示例中实现了LoaderManager.LoaderCallbacks这个接口，因此它传递this这个参数。

initLoader()方法能够确保加载器被初始化并激活。它有两种可能的结果：

1. 如果被ID标识的加载器已经存在，那么该加载器将被重新使用。
2. 如果被ID标识的加载器不存在，initLoader()将激发LoaderManager.LoaderCallbacks的onCreateLoader()方法。这能够实现并返回一个新加载器。

一般情况下，对LoaderManager.LoaderCallbacks的实现都与加载器紧密联系着一起，并且当加载器的状态变化时该方法将被调用。如果调用者在这个时候处于开始状态，并且那请求的加载器已经存在且已经产生了它的数据，然后系统将立即会调用onLoadFinished()方法(当initLoader()方法执行过程中)，所以我们必须为这个事情的发生做好准备。

我们需要注意initLoader()方法返回被创建的加载器，我们不必捕获对它的一个引用。因为LoaderManager会自动的管理加载器的生命。LoaderManager会在必要的时候开始和停止加载工作，并且会保持加载器的状态和加载器相关联的内容。这就意味着，我们很少直接与加载器们交互。我们大多数情况下当特殊的事件发生时去使用LoaderManager.LoaderCallbacks的回调方法去介入加载过程。

如果我们想放弃旧的数据，则应使用restartLoader()方法。

LoaderManager.LoaderCallbacks包含以下的方法：

1. onCreateLoader()：根据所给出的ID，初始化并返回一个新的加载器。
2. onLoadFinished()：当一个先前被创建的加载器完成了它的加载过程时被调用。
3. onLoaderReset()：当一个先前被创建的加载器被重置时被调用，然后使加载器的数据无效。

在onCreateLoader()方法中，能够使我们创建一个新的加载器，或者是实现我们自己的加载器。

在本示例中创建加载器的代码如下：



```
// 创建新的加载器
@Override
public Loader<Cursor> onCreateLoader(int arg0, Bundle arg1) {
    // TODO Auto-generated method stub
    // 为了查看信息，需要用到CursorLoader，也可以实现我们自己的加载器。
    CursorLoader cursorLoader = new CursorLoader(
        this,
        MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
        STORE_IMAGES,
        null,
        null,
        null);
    return cursorLoader;
}
```



在onLoadFinished()方法中，加载器一旦知道应用不再使用数据时，加载器将会释放数据。如果数据是一个从CursorLoader来的cursor，我们不应该调用它自己的close()方法，如果cursor被放置在CursorAdapter或是SimpleCursorAdapter中，我们应该使用它自己的swapCursor()方法以使旧的Cursor不被关闭。

在本示例中代码如下：

```
@Override
public void onLoadFinished(Loader<Cursor> arg0, Cursor cursor) {
    // TODO Auto-generated method stub
    // 使用swapCursor()方法，以使旧的游标不被关闭。
    simpleCursorAdapter.swapCursor(cursor);
}
```

在onLoaderReset()方法中，当一个先前被创建的加载器被重置时该方法会被调用，然后使加载器的数据无效。该回调方法让我们发现数据在什么时候将被释放以便我们删除对它的引用。通常我们实现swapCursor()方法，并给方法传递一个null参

数。

在本示例中代码如下：

```
@Override
public void onLoaderReset(Loader<Cursor> arg0) {
    // TODO Auto-generated method stub
    simpleCursorAdapter.swapCursor(null);
}
```

我通过Uri的形式查询到图片，代码如下：



```
Uri uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI.buildUpon().
    appendPath(Long.toString(id)).build();
FileUtil file = new FileUtil();
ContentResolver resolver = getContentResolver();

// 从Uri中读取图片资源
try {
    mContent = file.readInputStream(resolver.openInputStream(Uri.parse(uri.toString())));
    bitmap = file.getBitmapFromBytes(mContent, null);
    ivImageShow.setImageBitmap(bitmap);
} catch (Exception e) {
    // TODO: handle exception
    e.printStackTrace();
}
```



FileUtil.java文件主要是对图片Uri的处理过程。代码如下：



```
package com.device.photo;

import java.io.ByteArrayOutputStream;
```

```
import java.io.InputStream;

import android.graphics.Bitmap;
import android.graphics.BitmapFactory;

/**
 * 文件操作类
 * @Description: 文件操作类
 *
 * @FileName: FileUtil.java
 *
 * @Package com.device.photo
 *
 * @Author Hanyonglu
 *
 * @Date 2012-5-10 下午01:37:49
 *
 * @Version V1.0
 */
public class FileUtil {
    public FileUtil() {
        // TODO Auto-generated constructor stub
    }

    /**
     * InputStream to byte
     * @param inStream
     * @return
     * @throws Exception
     */
    public byte[] readInputStream(InputStream inStream) throws Exception {
        byte[] buffer = new byte[1024];
        int len = -1;
        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

        while ((len = inStream.read(buffer)) != -1) {
            outputStream.write(buffer, 0, len);
        }
    }
}
```



```
byte[] data = outputStream.toByteArray();
outputStream.close();
inputStream.close();

return data;
}

/**
 * Byte to bitmap
 * @param bytes
 * @param opts
 * @return
 */
public Bitmap getBitmapFromBytes(byte[] bytes, BitmapFactory.Options opts) {
    if (bytes != null){
        if (opts != null){
            return BitmapFactory.decodeByteArray(bytes, 0, bytes.length,opts);
        }
        else{
            return BitmapFactory.decodeByteArray(bytes, 0, bytes.length);
        }
    }

    return null;
}
}
```



在本示例中，我们不需要配置任何的权限就可以实现。

关于Android中使用加载器(Loader)来实现获取本机中的所有图片的实现过程及加载器(Loader)的使用大致上就介绍到这里了。

关于加载器更多的信息大家可以查阅官方帮助文档：<http://developer.android.com/guide/topics/fundamentals/loaders.html>

最后，希望转载的朋友能够尊重作者的劳动成果，加上转载地址：<http://www.cnblogs.com/hanyonglu/archive/2012/05/10/2494908.html>，谢谢。

Github：<https://github.com/hanyonglu/CursorLoader>

示例下载：[点击下载](#)。

完毕。^_^

分类：[【Android Develop】](#)



[Healtheon](#)

[关注 - 4](#)

[粉丝 - 910](#)

[+加关注](#)

6

0

« 上一篇：[Android实现导航菜单左右滑动效果](#)

» 下一篇：[OpenGL开发之旅基础知识介绍](#)

posted @ 2012-05-10 20:08 [Healtheon](#) 阅读(44617) 评论(15) [编辑](#) [收藏](#)

评论

#1楼 2012-05-11 16:42 | 明天陪你去看海

实例下载，点击下载，怎么没有啊？

[支持\(0\)](#) [反对\(0\)](#)

#2楼[楼主] 2012-05-11 17:01 | [Healtheon](#)

@ 明天陪你看海

哦，不好意思，我以为上传上去了。现在应该是可以了。

支持(0) 反对(0)

#3楼 2012-05-11 17:02 | 明天陪你看海

@ HarHar

现在可以了，谢谢！

支持(0) 反对(0)

#4楼 2012-05-12 09:39 | MudooT

很好啊 总也不做pad的开发 好多支持包里的东西都不会用呢 学习拉

支持(0) 反对(0)

#5楼 2012-06-06 13:57 | Pharaoh

关于这个包的详细信息请大家查看官方文档：<http://developer.android.com/sdk/compatibility-library.html>。

这个链接html的l没有包含在内，直接点开是404，需要手工将l加上。

支持(0) 反对(0)

#6楼[楼主] 2012-06-07 14:10 | Healtheon

@ Pharaoh

感谢你，我已经修改过了。

支持(0) 反对(0)

#7楼 2012-09-19 13:24 | 为梦奔跑

建议增加 dialog.setOnCancelListener(new OnCancelListener() {

@Override

public void onCancel(DialogInterface dialog) {

```
// TODO Auto-generated method stub
if(bitmap!=null)
{
    System.out.println("释放资源");
    bitmap.recycle();
}
}
});
和 public void onClick(View v) {
    dialog.dismiss();
    if(bitmap!=null)
    {
        bitmap.recycle();
    }
}
}
否则很容易发生内存溢出错误！！
```

支持(0) 反对(0)

#8楼[楼主] 2012-09-19 18:53 | Healtion

@ 为梦奔跑
谢谢您的建议。

支持(0) 反对(0)

#9楼 2013-07-01 17:01 | chenwj

LZ，怎么运行不了？

支持(0) 反对(0)

#10楼 2013-07-01 17:18 | chenwj

LZ,android4.0以上的是不是不能跑啊

支持(0) 反对(0)

#11楼[楼主] 2013-07-01 19:09 | Healtion

@ chenwj

可以运行的起来，我这边是可以的啊。你可以调试下，看看是什么问题.....

支持(0) 反对(0)

#12楼 2014-08-08 10:54 | FallCanBeFree

下载看一下，谢谢楼主分享

支持(0) 反对(0)

#13楼 2015-06-04 11:16 | 奇v遇

@ Healtheon

楼主 下不了啊

支持(0) 反对(0)

#14楼 2016-01-21 16:04 | 天空无痕

安装到手机上直接打不开?停止运行？

支持(0) 反对(0)

#15楼 2016-04-07 10:11 | hok

你好，我想问一下，这个读取的图片包括 系统data目录下的吗？还是仅仅是sdcard目录下的？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】搭建微信小程序 就选腾讯云

【推荐】报表开发有捷径：快速设计轻松集成，数据可视化和交互



最新IT新闻:

- 一图看懂史上最大规模的云栖大会 马云或再展歌喉
 - Apple Watch Series 3续航如何：LTE下不乐观
 - 增加介绍与商店连接 谷歌首页小幅改版
 - iOS 11：无电源按钮的情况如何关闭iPhone
 - 人类“换头术”现在就是一场炒作 但医学界不会放弃研究
- » 更多新闻...



最新知识库文章:

- 实用VPC虚拟私有云设计原则
 - 如何阅读计算机科学类的书
 - Google 及其云智慧
 - 做到这一点，你也可以成为优秀的程序员
 - 写给立志做码农的大学生
- » 更多知识库文章...