[как хранить изображение в SQLite базы данных](http://ru.androids.help/q13072)

В моем приложении я загрузки изображения из галереи и образ, который я загрузил хотите сохранить в database.But как для хранения растровых в базе данных я уверен, преобразование растрового изображения в строке и сохранение в базе данных, но в то время как retriving на, как она есть строка не умею назначить эту строку в ImageView .

Imageupload.class :

public class Imageupload12 extends Activity {

Button buttonLoadImage;

ImageView targetImage;

int i=0;

Database database=new Database(this);

String i1;

String img;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main5);

buttonLoadImage = (Button)findViewById(R.id.loadimage);

targetImage = (ImageView)findViewById(R.id.targetimage);

Bundle b=getIntent().getExtras();

if(b!=null)

{

img=b.getString("image");

targetImage2.setImageURI("image");

//i am getting error as i cant assign string to imageview.

}

buttonLoadImage.setOnClickListener(new Button.OnClickListener(){

public void onClick(View arg0) {

// TODO Auto-generated method stub

Intent intent = new Intent(Intent.ACTION\_PICK,

android.provider.MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);

Log.i("photo",""+intent);

startActivityForResult(intent, i);

i=i+1;

}});

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

// TODO Auto-generated method stub

super.onActivityResult(requestCode, resultCode, data);

switch(requestCode) {

case 0:

if (resultCode == RESULT\_OK){

Uri targetUri = data.getData();

// textTargetUri.setText(targetUri.toString());

Bitmap bitmap;

try {

bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(targetUri));

targetImage.setImageBitmap(bitmap);

i1=bitmap.toString();

Log.i("firstimage........",""+i1);

targetImage.setVisibility(0);

SQLiteDatabase db=database.getWritableDatabase();

db.execSQL("INSERT INTO UPLOAD VALUES('"+i1+"');");

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

break;

}

}

}

Image.class:

public class Image extends Activity {

Database database=new Database(this);

static EfficientAdapter adapter, adapter1;

static ListView lv1;

static SQLiteDatabase db;

static EfficientAdapter adp;

static Cursor c1;

static Vector<String>IMAGE=new Vector<String>();

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

db=database.getReadableDatabase();

c1=db.rawQuery("select \* from UPLOAD;", null);

if (c1.moveToFirst()) {

do {

IMAGE.add(c1.getString(0).toString());

} while (c1.moveToNext());

c1.close();

}

lv1=(ListView)findViewById(R.id.List);

adapter=new EfficientAdapter(this);

lv1.setAdapter(adapter);

ImageView add=(ImageView)findViewById(R.id.imv1a);

add.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

IMAGE.clear();

Intent i=new Intent(Image.this,Imageupload12.class);

startActivity(i);

}

});

}

private static class EfficientAdapter extends BaseAdapter{

// protected final Context Context = null;

protected LayoutInflater mLayoutInflater;

AlertDialog.Builder aBuilder;

public EfficientAdapter(Context context) {

// TODO Auto-generated constructor stub

mLayoutInflater=LayoutInflater.from(context);

}

@Override

public int getCount() {

// TODO Auto-generated method stub

return IMAGE.size();

}

@Override

public Object getItem(int position) {

// TODO Auto-generated method stub

return position;

}

@Override

public long getItemId(int position) {

// TODO Auto-generated method stub

return position;

}

@Override

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

// TODO Auto-generated method stub

final ViewHolder mVHolder;

if(convertView == null){

convertView=mLayoutInflater.inflate(R.layout.pjtlistdetails, parent, false);

mVHolder=new ViewHolder();

mVHolder.t1=(TextView)convertView.findViewById(R.id.pjtdetails);

mVHolder.time=(TextView)convertView.findViewById(R.id.name);

mVHolder.imv=(ImageButton)convertView.findViewById(R.id.editic);

mVHolder.imvd=(ImageView)convertView.findViewById(R.id.delete);

mVHolder.imvf=(ImageView)convertView.findViewById(R.id.fwd);

mVHolder.imv.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String img=IMAGE.elementAt(position);

Log.i("image...",""+img);

Context ctx=v.getContext();

Intent myIntent = new Intent();

ctx = v.getContext();

myIntent.setClass(ctx, Imageupload12.class);

myIntent.putExtra("image", img);

ctx.startActivity(myIntent);

IMAGE.clear();

}

});

static class ViewHolder{

ImageButton imv;

ImageView imvd,imvf;

}

Top 5 ответ

**1**[Sumant](http://stackoverflow.com/users/630313/)@2012-02-20 07:47:52

Чтобы сохранить любое изображение в SQLite базы данных необходимо сохранить это изображение в массив байтов, а не строки. Преобразование этот образ в массив байтов и усилителя ; сохранить этот байт [] к БД . Во время определения, что изображение, которое вы получите байт [] преобразовать этот байт [ ], чтобы растровые, с помощью которого вы получите оригинальное изображение.

**2**[Shahzad Imam](http://stackoverflow.com/users/1109930/)@2012-02-20 07:54:00

Вы должны просто хранить образ пути к sqlite.It не мудро, чтобы сохранить двоичные данные в SQLite. Вы должны следить за этим http://stackoverflow.com/questions/9273008/android-save-images-to-sqlite-or-sdcard-or-memory/9273045#9273045 страница, чтобы очистить ваши сомнения

**3**[Jram](http://stackoverflow.com/users/1172865/)@2012-02-20 08:12:34

Вы должны использовать " капля ", чтобы сохранить изображение.

например: для хранения образа и БД

public void insertImg(int id, Bitmap img ) {

byte[] data = getBitmapAsByteArray(img); // this is a function

insertStatement\_logo.bindLong(1, id);

insertStatement\_logo.bindBlob(2, data);

insertStatement\_logo.executeInsert();

insertStatement\_logo.clearBindings() ;

}

public static byte[] getBitmapAsByteArray(Bitmap bitmap) {

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

bitmap.compress(CompressFormat.PNG, 0, outputStream);

return outputStream.toByteArray();

}

для получения изображения из БД

public Bitmap getImage(int i){

String qu = "select img from table where feedid=" + i ;

Cursor cur = db.rawQuery(qu, null);

if (cur.moveToFirst()){

byte[] imgByte = cur.getBlob(0);

cur.close();

return BitmapFactory.decodeByteArray(imgByte, 0, imgByte.length);

}

if (cur != null && !cur.isClosed()) {

cur.close();

}

return null ;

}

**4**[user2288580](http://stackoverflow.com/users/2288580/)@2014-03-05 03:36:06

Я считаю, чтолучший способ хранения изображения на SQLLite базы данных, чтобы использовать базовый алгоритм 64 . который преобразует изображение в обычный текст и обратно. Вы можете скачать полный пример проекта Android на : http://developersfound.com/Base64FromStream.zip . Эта программа не сохраняет образ, но он делает конвертировать изображения из изображения в текст и обратно.

Воткласс:

package com.example.TestProject;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.util.Base64;

import android.util.Log;

import java.io.\*;

import java.net.URL;

import java.net.URLConnection;

import java.nio.channels.FileChannel;

public class Base64CODEC {

private int IO\_BUFFER\_SIZE = 64;

//private int IO\_BUFFER\_SIZE = 8192;

private URL urlObject = null;

private URLConnection myConn = null;

ByteArrayOutputStream os = null;

public void Base64CODEC() {}

public Bitmap Base64ImageFromURL(String url) {

Bitmap bitmap = null;

InputStream in = null;

BufferedOutputStream out = null;

try {

urlObject = new URL(url);

myConn = urlObject.openConnection();

in = myConn.getInputStream();

final ByteArrayOutputStream dataStream = new ByteArrayOutputStream();

out = new BufferedOutputStream(dataStream, IO\_BUFFER\_SIZE);

copyCompletely(in, out);

final byte[] data = dataStream.toByteArray();

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

bitmap = BitmapFactory.decodeByteArray(data, 0, data.length, options);

} catch (IOException e) {

Log.e("TAG", "Could not load Bitmap from: " + url);

} finally {

//closeStream(in);

try {

in.close();

} catch (IOException e) {

e.printStackTrace(); //To change body of catch statement use File | Settings | File Templates.

}

//closeStream(out);

try {

out.close();

} catch (IOException e) {

e.printStackTrace(); //To change body of catch statement use File | Settings | File Templates.

}

}

return bitmap;

}

private void copyCompletely(InputStream input, OutputStream output) throws IOException {

// if both are file streams, use channel IO

if ((output instanceof FileOutputStream) && (input instanceof FileInputStream)) {

try {

FileChannel target = ((FileOutputStream) output).getChannel();

FileChannel source = ((FileInputStream) input).getChannel();

source.transferTo(0, Integer.MAX\_VALUE, target);

source.close();

target.close();

return;

} catch (Exception e) { /\* failover to byte stream version \*/

}

}

byte[] buf = new byte[8192];

while (true) {

int length = input.read(buf);

if (length < 0)

break;

output.write(buf, 0, length);

}

try {

input.close();

} catch (IOException ignore) {

}

try {

output.close();

} catch (IOException ignore) {}

}

public String convertToBase64(Bitmap bitmap) {

ByteArrayOutputStream os = new ByteArrayOutputStream();

bitmap.compress(Bitmap.CompressFormat.PNG,100,os);

byte[] byteArray = os.toByteArray();

return Base64.encodeToString(byteArray, 0);

}

public Bitmap convertToBitmap(String base64String) {

byte[] decodedString = Base64.decode(base64String, Base64.DEFAULT);

Bitmap bitmapResult = BitmapFactory.decodeByteArray(decodedString, 0, decodedString.length);

return bitmapResult;

}

}

А вотосновная деятельность, которая использует класс :

package com.example.TestProject;

import android.app.Activity;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.view.View;

import android.widget.ImageView;

public class MainActivity extends Activity implements Runnable {

private Thread thread = null;

private Bitmap bitmap = null;

private Base64CODEC base64CODEC = null;

private ImageView imgViewSource = null;

private ImageView imgViewDestination = null;

private boolean isSourceImageVisible = false;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

}

public void CmdLoadImage\_Click(View view) {

try {

if(isSourceImageVisible == true) {

imgViewSource.setImageBitmap(null);

imgViewDestination.setImageBitmap(null);

isSourceImageVisible = false;

}

else {

base64CODEC = new Base64CODEC();

thread = new Thread(this);

thread.start();

}

}

catch (NullPointerException e) {}

}

public void CmdEncodeImage\_Click(View view) {

Base64CODEC base64CODEC = new Base64CODEC();

try {

String base64String = base64CODEC.convertToBase64(bitmap);

imgViewDestination = (ImageView) findViewById(R.id.imgViewDestination);

Bitmap imgViewDestinationBitmap = base64CODEC.convertToBitmap(base64String);

imgViewDestination.setImageBitmap(imgViewDestinationBitmap);

}

catch (NullPointerException e) {

//

}

}

@Override

public void run() {

bitmap = base64CODEC.Base64ImageFromURL("http://developersfound.com/me.png");

handler.sendEmptyMessage(0);

}

private Handler handler = new Handler() {

@Override

public void handleMessage(Message msg) {

imgViewSource = (ImageView) findViewById(R.id.imgViewSource);

imgViewSource.setImageBitmap(bitmap);

isSourceImageVisible = true;

thread = null;

}

};

}

* [Естьспособ проверить, если Android устройство поддерживает OpenGL ES 2.0 ? ?](http://ru.androids.help/q13388)