Сохранение картинок в SQLITE

Если вы действительно храните это по какой-либо причине, вы также можете попытаться закодировать изображение как Base64 Stringи сохранить его в своей БД.

Android имеет [Base64 class](http://developer.android.com/reference/android/util/Base64.html) для этого. Попробуйте использовать следующий фрагмент кода для кодирования:

ByteArrayOutputStream baos = new ByteArrayOutputStream();

bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos); // Could be Bitmap.CompressFormat.PNG or Bitmap.CompressFormat.WEBP

byte[] bai = baos.toByteArray();

String base64Image = Base64.encodeToString(bai, Base64.DEFAULT);

// Call your method to save this string on the DB here.

И вам придется декодировать его, попробуйте следующее:

byte[] data = Base64.decode(base64Image, Base64.DEFAULT);

Bitmap bm;

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

opt.inMutable = true;

bm = BitmapFactory.decodeByteArray(data, 0, data.length, opt);

// Now do whatever you want with the Bitmap.

Вы можете увидеть документы для Bitmapкласса [здесь](http://developer.android.com/reference/android/graphics/Bitmap.html) .

profile\_images.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent photoPickerIntent = new Intent(Intent.ACTION\_PICK);

photoPickerIntent.setType("image/\*");

startActivityForResult(photoPickerIntent, SELECT\_PHOTO);

}

});

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

super.onActivityResult(requestCode, resultCode, imageReturnedIntent);

switch(requestCode) {

case SELECT\_PHOTO:

if(resultCode == RESULT\_OK){

Uri selectedImage = imageReturnedIntent.getData();

try {

Bitmap bmp = decodeUri(selectedImage);

profile\_images.setImageBitmap(bmp);

ByteArrayOutputStream stream = new ByteArrayOutputStream();

bmp.compress(Bitmap.CompressFormat.JPEG, 100, stream);

byte[] byteArray = stream.toByteArray();

insertUser(byteArray);

}catch (Exception e){

e.printStackTrace();

}

}

}

}

public void insertUser(byte[] logoImage ){

SQLiteDatabase db = dbs.getWritableDatabase();

String delSql = "DELETE FROM Image";

SQLiteStatement delStmt = db.compileStatement(delSql);

delStmt.execute();

String sql = "INSERT INTO Image (CODE,Img) VALUES(?,?)";

SQLiteStatement insertStmt = db.compileStatement(sql);

insertStmt.clearBindings();

insertStmt.bindLong(1, 1);

insertStmt.bindBlob(2,logoImage);

// insertStmt.bindBlob(3, this.accImage);

insertStmt.executeInsert();

db.close();

}

public Bitmap getCurrentBitmap() {

SQLiteDatabase db = dbs.getWritableDatabase();

String sql = "SELECT \* FROM Image";

Cursor cursor = db.rawQuery(sql, new String[]{});

Bitmap bmp=null;

if(cursor.moveToFirst()){

//this.accId = cursor.getInt(0);

// this.accName = cursor.getString(1);

byte[] logoImage = cursor.getBlob(1);

bmp = BitmapFactory.decodeByteArray(logoImage, 0, logoImage.length);

}

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

cursor.close();

}

db.close();

if(cursor.getCount() == 0){

return null;

} else {

return bmp;

}

}

private Bitmap decodeUri(Uri selectedImage) throws FileNotFoundException {

// Decode image size

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

o.inJustDecodeBounds = true;

BitmapFactory.decodeStream(getContentResolver().openInputStream(selectedImage), null, o);

// The new size we want to scale to

final int REQUIRED\_SIZE = 140;

// Find the correct scale value. It should be the power of 2.

int width\_tmp = o.outWidth, height\_tmp = o.outHeight;

int scale = 1;

while (true) {

if (width\_tmp / 2 < REQUIRED\_SIZE

|| height\_tmp / 2 < REQUIRED\_SIZE) {

break;

}

width\_tmp /= 2;

height\_tmp /= 2;

scale \*= 2;

}

// Decode with inSampleSize

BitmapFactory.Options o2 = new BitmapFactory.Options();

o2.inSampleSize = scale;

return BitmapFactory.decodeStream(getContentResolver().openInputStream(selectedImage), null, o2);

}

you can do this as above given methods.