I want to store an image(from url) into a sqlite database.

For that I use:

db = new DataBase(getApplicationContext());

URL url = new URL("http://sree.cc/wp-content/uploads/schogini\_team.png");

URLConnection ucon = url.openConnection();

InputStream is = ucon.getInputStream();

BufferedInputStream bis = new BufferedInputStream(is,128);

ByteArrayBuffer barb= new ByteArrayBuffer(128);

int current = 0;

while ((current = bis.read()) != -1) {

barb.append((byte) current);

}

ContentValues filedata= new ContentValues();

filedata.put(DataBase.IMG\_SRC,barb.toByteArray());

db.insert(DataBase.Table\_Img, null, filedata);

In the Insert():

public void insert(String tableImg, Object object,

ContentValues dataToInsert) {

// TODO Auto-generated method stub

String sql = "INSERT INTO "+tableImg+" ("+ID+","+IMG\_SRC+") " +

"VALUES ('"+1+"','"+dataToInsert+"')";

db.execSQL(sql);

}

For the retrieval of image:

Cursor cursor = db.selectDataToShow(DataBase.Table\_Img, DataBase.IMG\_SRC);

byte[] imageByteArray=cursor.getBlob(cursor.getColumnIndex(DataBase.IMG\_SRC));

cursor.close();

ByteArrayInputStream imageStream = new ByteArrayInputStream(imageByteArray);

Bitmap theImage = BitmapFactory.decodeStream(imageStream);

System.out.println(">>>>>>>>>>>>>>>>>>>>>> "+theImage);

So here I got null.

And in my database the value of image stored as: Image=[B@43e5ac48]

byte[] logoImage = getLogoImage(IMAGEURL);

private byte[] getLogoImage(String url){

try {

URL imageUrl = new URL(url);

URLConnection ucon = imageUrl.openConnection();

InputStream is = ucon.getInputStream();

BufferedInputStream bis = new BufferedInputStream(is);

ByteArrayBuffer baf = new ByteArrayBuffer(500);

int current = 0;

while ((current = bis.read()) != -1) {

baf.append((byte) current);

}

return baf.toByteArray();

} catch (Exception e) {

Log.d("ImageManager", "Error: " + e.toString());

}

return null;

}

To save the image to db i used this code.

public void insertUser(){

SQLiteDatabase db = dbHelper.getWritableDatabase();

String delSql = "DELETE FROM ACCOUNTS";

SQLiteStatement delStmt = db.compileStatement(delSql);

delStmt.execute();

String sql = "INSERT INTO ACCOUNTS (account\_id,account\_name,account\_image) VALUES(?,?,?)";

SQLiteStatement insertStmt = db.compileStatement(sql);

insertStmt.clearBindings();

insertStmt.bindString(1, Integer.toString(this.accId));

insertStmt.bindString(2,this.accName);

insertStmt.bindBlob(3, this.accImage);

insertStmt.executeInsert();

db.close();

}

To retrieve the image back this is code i used.

public Account getCurrentAccount() {

SQLiteDatabase db = dbHelper.getWritableDatabase();

String sql = "SELECT \* FROM ACCOUNTS";

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

if(cursor.moveToFirst()){

this.accId = cursor.getInt(0);

this.accName = cursor.getString(1);

this.accImage = cursor.getBlob(2);

}

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

cursor.close();

}

db.close();

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

return null;

} else {

return this;

}

}

Finally to load this image to a imageview

logoImage.setImageBitmap(BitmapFactory.decodeByteArray( currentAccount.accImage,

0,currentAccount.accImage.length));

[share](https://stackoverflow.com/a/7331698)[improve this answer](https://stackoverflow.com/posts/7331698/edit)

in the DBAdaper i.e Data Base helper class declare the table like this

private static final String USERDETAILS=

"create table userdetails(usersno integer primary key autoincrement,userid text not null ,username text not null,password text not null,photo BLOB,visibility text not null);";

insert the values like this,

first convert the images as byte[]

ByteArrayOutputStream baos = new ByteArrayOutputStream();

Bitmap bitmap = ((BitmapDrawable)getResources().getDrawable(R.drawable.common)).getBitmap();

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

byte[] photo = baos.toByteArray();

db.insertUserDetails(value1,value2, value3, photo,value2);

in DEAdaper class

public long insertUserDetails(String uname,String userid, String pass, byte[] photo,String visibility)

{

ContentValues initialValues = new ContentValues();

initialValues.put("username", uname);

initialValues.put("userid",userid);

initialValues.put("password", pass);

initialValues.put("photo",photo);

initialValues.put("visibility",visibility);

return db.insert("userdetails", null, initialValues);

}

retrieve the image as follows

Cursor cur=your query;

while(cur.moveToNext())

{

byte[] photo=cur.getBlob(index of blob cloumn);

}

ByteArrayInputStream imageStream = new ByteArrayInputStream(photo);

Bitmap theImage= BitmapFactory.decodeStream(imageStream);

In insert()

public void insert(String tableImg, Object object,

ContentValues dataToInsert) {

db.insert(tablename, null, dataToInsert);

}

Hope it helps you.

[share](https://stackoverflow.com/a/7331644)[improve this answer](https://stackoverflow.com/posts/7331644/edit)

answered Sep 7 '11 at 9:34

[[https://www.gravatar.com/avatar/9a5e2a428761bdd31e9314b7adafa288?s=32&d=identicon&r=PG](https://stackoverflow.com/users/385343/nishant-shah)](https://stackoverflow.com/users/385343/nishant-shah)

[Nishant Shah](https://stackoverflow.com/users/385343/nishant-shah)

**1,715**51928

add a comment

up vote0down vote

you may also want to encode and decode to/from base64

function uncompress(str:String):ByteArray {

import mx.utils.Base64Decoder;

var dec:Base64Decoder = new Base64Decoder();

dec.decode(str);

var newByteArr:ByteArray=dec.toByteArray();

return newByteArr;

}

// Compress a ByteArray into a Base64 String.

function compress(bytes:ByteArray):String {

import mx.utils.Base64Decoder; //Transform String in a ByteArray.

import mx.utils.Base64Encoder; //Transform ByteArray in a readable string.

var enc:Base64Encoder = new Base64Encoder();

enc.encodeBytes(bytes);

return enc.drain().split("\n").join("");

}

[share](https://stackoverflow.com/a/17858022)[improve this answer](https://stackoverflow.com/posts/17858022/edit)

answered Jul 25 '13 at 12:27

[[https://www.gravatar.com/avatar/d799a44dfb6dc3cb99d4cbb41436a47c?s=32&d=identicon&r=PG](https://stackoverflow.com/users/2433181/johnny)](https://stackoverflow.com/users/2433181/johnny)

[johnny](https://stackoverflow.com/users/2433181/johnny)

**1,006**526

add a comment

up vote0down vote

for a ionic project

var imgURI = "";

var imgBBDD = ""; //sqllite for save into

function takepicture() {

var options = {

quality : 75,

destinationType : Camera.DestinationType.DATA\_URL,

sourceType : Camera.PictureSourceType.CAMERA,

allowEdit : true,

encodingType: Camera.EncodingType.JPEG,

targetWidth: 300,

targetHeight: 300,

popoverOptions: CameraPopoverOptions,

saveToPhotoAlbum: false

};

$cordovaCamera.getPicture(options).then(function(imageData) {

imgURI = "data:image/jpeg;base64," + imageData;

imgBBDD = imageData;

}, function(err) {

// An error occured. Show a message to the user

});

}

And now we put imgBBDD into SqlLite

function saveImage = function (theId, theimage){

var insertQuery = "INSERT INTO images(id, image) VALUES("+theId+", '"+theimage+"');"

console.log('>>>>>>>');

DDBB.SelectQuery(insertQuery)

.then( function(result) {

console.log("Image saved");

})

.catch( function(err)

{

deferred.resolve(err);

return cb(err);

});

}

A server side (php)

$request = file\_get\_contents("php://input"); // gets the raw data

$dades = json\_decode($request,true); // true for return as array

if($dades==""){

$array = array();

$array['error'] = -1;

$array['descError'] = "Error when get the file";

$array['logError'] = '';

echo json\_encode($array);

exit;

}

//send the image again to the client

header('Content-Type: image/jpeg');

echo '';

byte[] byteArray = rs.getBytes("columnname");

Bitmap bm = BitmapFactory.decodeByteArray(byteArray, 0 ,byteArray.length);

Cursor cur=your query;

while(cur.moveToNext())

{

byte[] photo=cur.getBlob(index of blob cloumn);

}

convert the byte[] into image

ByteArrayInputStream imageStream = new ByteArrayInputStream(photo);

Bitmap theImage= BitmapFactory.decodeStream(imageStream);