Backing up, importing and restoring databases on Android

14 September 2010 · Horatio Caine · Android, Programming - Tags: android, backup, code, database, example, file, import, load, restore, save, source, store, valid

20 comments

If your Android application has a database and you want your users to be able to backup the database and restore it as they see fit, you'll need to mess about with database files.

Below is a class called DbExportImport. Once you've set it up with the correct values, all you need to do is call **exportDb()**, **importDb()** or **restoreDb()** from your application to perform the necessary operations.

This is also useful as a temporary measure when changing your package name or key for application signing, as your application will be newly installed and you will lose your database.

I have only left in a few comments, so for any clarification, leave a comment and I'll make it a little clearer.

```
package testCode;
```

Recent Posts

Four seconds left to live – what went through my mind

How to stop the throttling on your Virgin Media broadband connection

Enabling command history in terminal programs like Sicstus

Snippets – Part I

First line disappears from Eclipse console output

Applying for an IBM industrial placement – a complete guide

When Eclipse won't generate/locate R.java

My job (placement) offer from IBM!

SQLite timestamp comparison causes SQLiteException or error

Backing up, importing and restoring databases on Android

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.channels.FileChannel;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteException;
import android.os.Environment;
import android.text.format.DateFormat;
import android.util.Log;

public class DbExportImport {

        public static final String TAG = DbExportImport.class.getName();

        /** Directory that files are to be read from and written to **/
        protected static final File DATABASE_DIRECTORY =
                new File(Environment.getExternalStorageDirectory(),"MyDirectory");

        /** File path of Db to be imported **/
        protected static final File IMPORT_FILE =
                new File(DATABASE_DIRECTORY,"MyDb.db");

        public static final String PACKAGE_NAME = "com.example.app";
        public static final String DATABASE_NAME = "example.db";
        public static final String DATABASE_TABLE = "entryTable";

        /** Contains: /data/data/com.example.app/databases/example.db **/
        private static final File DATA_DIRECTORY_DATABASE =
                        new File(Environment.getDataDirectory() +
                        "/data/" + PACKAGE_NAME +
                        "/databases/" + DATABASE_NAME );

        /** Saves the application database to the
         * export directory under MyDb.db **/
        protected static  boolean exportDb(){
                if( ! SdIsPresent() ) return false;
```

```
        File dbFile = DATA_DIRECTORY_DATABASE;
        String filename = "MyDb.db";

        File exportDir = DATABASE_DIRECTORY;
        File file = new File(exportDir, filename);

        if (!exportDir.exists()) {
            exportDir.mkdirs();
        }

        try {
            file.createNewFile();
            copyFile(dbFile, file);
            return true;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
    }

    /** Replaces current database with the IMPORT_FILE if
     * import database is valid and of the correct type **/
    protected static boolean restoreDb(){
        if( ! SdIsPresent() ) return false;

        File exportFile = DATA_DIRECTORY_DATABASE;
        File importFile = IMPORT_FILE;

        if( ! checkDbIsValid(importFile) ) return false;

        if (!importFile.exists()) {
            Log.d(TAG, "File does not exist");
            return false;
        }

        try {
            exportFile.createNewFile();
            copyFile(importFile, exportFile);
            return true;
        } catch (IOException e) {
            e.printStackTrace();
            return false;
```

```
        }
    }

    /** Imports the file at IMPORT_FILE **/
    protected static boolean importIntoDb(Context ctx){
        if( ! SdIsPresent() ) return false;

        File importFile = IMPORT_FILE;

        if( ! checkDbIsValid(importFile) ) return false;

        try{
            SQLiteDatabase sqlDb = SQLiteDatabase.openDatabase
                    (importFile.getPath(), null, SQLiteDatabase

            Cursor cursor = sqlDb.query(true, DATABASE_TABLE,
                    null, null, null, null, null, null, null
               );

            DbAdapter dbAdapter = new DbAdapter(ctx);
            dbAdapter.open();

            final int titleColumn = cursor.getColumnIndexOrThrow("title
            final int timestampColumn = cursor.getColumnIndexOrThrow("t

            // Adds all items in cursor to current database
            cursor.moveToPosition(-1);
            while(cursor.moveToNext()){
                dbAdapter.createQuote(
                        cursor.getString(titleColumn),
                        cursor.getString(timestampColumn)
                );
            }

            sqlDb.close();
            cursor.close();
            dbAdapter.close();
        } catch( Exception e ){
            e.printStackTrace();
            return false;
        }
```

```
            return true;
        }

        /** Given an SQLite database file, this checks if the file
         * is a valid SQLite database and that it contains all the
         * columns represented by DbAdapter.ALL_COLUMN_KEYS **/
        protected static boolean checkDbIsValid( File db ){
            try{
                SQLiteDatabase sqlDb = SQLiteDatabase.openDatabase
                        (db.getPath(), null, SQLiteDatabase.OPEN_READONLY);

                Cursor cursor = sqlDb.query(true, DATABASE_TABLE,
                null, null, null, null, null, null, null
            );

                // ALL_COLUMN_KEYS should be an array of keys of essential
                // Throws exception if any column is missing
                for( String s : DbAdapter.ALL_COLUMN_KEYS ){
                        cursor.getColumnIndexOrThrow(s);
                }

                sqlDb.close();
                cursor.close();
            } catch( IllegalArgumentException e ) {
                Log.d(TAG, "Database valid but not the right type");
                e.printStackTrace();
                return false;
            } catch( SQLiteException e ) {
                Log.d(TAG, "Database file is invalid.");
                e.printStackTrace();
                return false;
            } catch( Exception e){
                Log.d(TAG, "checkDbIsValid encountered an exception");
                e.printStackTrace();
                return false;
            }

            return true;
        }

        private static void copyFile(File src, File dst) throws IOException {
```

```
                FileChannel inChannel = new FileInputStream(src).getChannel();
                FileChannel outChannel = new FileOutputStream(dst).getChannel();
                try {
                        inChannel.transferTo(0, inChannel.size(), outChannel);
                } finally {
                        if (inChannel != null)
                                inChannel.close();
                        if (outChannel != null)
                                outChannel.close();
                }
        }

        /** Returns whether an SD card is present and writable **/
        public static boolean SdIsPresent() {
                return Environment.getExternalStorageState().equals(
                                Environment.MEDIA_MOUNTED);
        }
}
```


20 Responses to “Backing up, importing and restoring databases on Android”

Otis on February 18th, 2011 at 2:13 am #

Would you please post the DbAdapter source code to me?
I have some problems about use db in adapter. It looks helpful.

Reply

Horatio Caine on February 18th, 2011 at 10:00 am #

@Otis
Sorry, Otis. DbAdapter is code specially written for my own application so I cannot put it up.
However, you should download Android's sample 'Notes' application, which has a DbAdapter class with functionality that you may find useful.
Reply

Otis on February 21st, 2011 at 3:19 am #

Thanks a lot@Horatio Caine
Reply

Ramiro Kousonsavath on March 18th, 2011 at 3:37 pm #

quite intriguing post
Reply

Marcos Plasencia on March 28th, 2011 at 11:43 am #

I will definitely recommend this to my friends
Reply

peter on April 1st, 2011 at 1:10 pm #

Hi Horatio so useful post!!
Let me ask you a doubt! maybe it's a stupid question , but I am pretty new in android.

I can't see clear this: " dbAdapter.createQuote"
What does the createQuote do? I am looking how to write it inside of my DbCreate, but I don't know how, could you give an example maybe?
Thanks

Reply

Horatio Caine on April 2nd, 2011 at 11:49 am #

@peter
Hi Peter.
I'm glad you found the post useful. I can't give you any sample code for DbAdapter, since it's part of my own application, but as I've mentioned in comments to Otis above, the Notes application contains the information you need.
Just download the sample code and have a look at their own DbAdapter class. Usually you just extend SQLiteOpenHelper and add your own specific functionality.
Good luck,
Horatio

Reply

Gautam on February 24th, 2012 at 9:34 am #

Hi Horatio,
Your code is really helpful, but I cannot figure out how your import functionality works. Does it import only the .db file? If that is so how should I get the row wise data of the tables in the .db file? Do i need to create the same tables again so that I can fill the data from the imported .db file??
Please do reply for my confusion…Really need your help..
Thanking you,
Gautam.

Reply

Horatio Caine on February 24th, 2012 at 6:48 pm #

Hi Gautam, thanks for stopping by.
The import method only demonstrates how to take a database file path, create a DB object from it and use that object to query the data within. I haven't covered how to put the data into your own table, although that should be fairly straightforward once you write some methods which take in data and insert a row into your existing database. There is code showing how to enter data into tables in the Notes sample application on the Android dev pages.
Horatio

Reply

Gautam on February 25th, 2012 at 6:27 am #

Hi Horatio,
Yes, I was able to figure out how to populate the table within the my database from the imported .db file. Your CreateQuote() kind of confused me. hehe… thanks for the code. I'll surely look for other important posts that you have in your blog. Nice to meet you Friend!!!
Thanking you,
Gautam

Reply

Horatio Caine on February 25th, 2012 at 9:04 am #

Yes, I apologise for not making that clearer it's just a utility method to populate my existing database. I'm glad I could help :-)

Reply

Gautam on February 25th, 2012 at 9:59 am #

hehehehehe…..Thanks!!

nischal on May 8th, 2012 at 3:40 am #

how to implement DBadapter.createquote();

Reply

Horatio Caine on May 8th, 2012 at 9:13 am #

@nischal
Please see my comments to Peter and Otis above

Reply

Cory on June 14th, 2012 at 2:08 pm #

Hello. I found this very helpful but is there a way I can back up the Database file to some sort of cloud storage as opposed to external hardware or another folder on the phone? Thanks.

Reply

Horatio Caine on June 16th, 2012 at 9:24 pm #

@Cory

That is not really within the scope of this post. Once you have your database file, you can do whatever you like with it. If you want to back it up to the web, you can write some separate code to upload the file once you've produced it.

Reply

paul on July 17th, 2012 at 6:20 pm #

this code will be helpful to me since am making a backup application as my project. Thanks alot.
I will recommend this blog to my pals

Reply

Márcio Las on August 15th, 2012 at 11:48 pm #

Hello! I am a beginner in Android, I implemented your code and testing always results in a DB file empty (0 bytes). Can you tell me why? Thanks in advance for your attention.

Reply

Rock on October 3rd, 2012 at 12:35 pm #

Thanks a lot @Horatio Caine….
The code is really useful and easy to understand.
It really worked for me.
Keep on posting sample code.

Reply

Mark on October 5th, 2012 at 5:50 pm #

@Horatio… Thank You!
This worked perfectly for me. I'm learning Java as I go developing an Android app. I was able to understand each line of code and this brought me one step closer to being able to actually program and not just copy and paste.
Reply

Leave Your Response

Name (optional)

E-mail (w ill not be published) (optional)

Website (optional)

Submit Comment

☐ Notify me of follow -up comments by email.

☐ Notify me of new posts by email.