LEARNING android-sqlite

#android-sqlite

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: android-sqlite

It is an unofficial and free android-sqlite ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-sqlite.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with android-sqlite

Remarks

The SQLite library itself has only a C API; to make it accessible from Java, the Android framework wraps this with the android.database.sqlite package. The most important classes are SQLiteDatabase and SQLiteOpenHelper.

The android.database package contains the database-related parts of the framework that are not SQLite specific.

Versions

API Level

API Level	Platform Version	Name	Release Date
1	1.0	Base	2008-09-23
5	2.0	Eclair	2009-10-26
8	2.2	Froyo	2010-05-20
11	3.0	Honeycomb	2011-02-22
16	4.1	Jelly Bean	2012-07-09
19	4.4	Kitkat	2013-10-31
23	6.0	Marshmallow	2015-10-05

Examples

Basic usage

To include a database in your app, you typically derive a class from SQLiteOpenHelper:

```
public class HelloDBHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 1;
    private static final int DATABASE_NAME = "hello";

    HelloDBHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE ...");
        ...
    }
}
```

This helper class is responsible for opening (and creating/updating, if needed) the database. Use it to get an SQLiteDatabase object to access the data:

```
SQLiteDatabase db = helper.getReadableDatabase();
Cursor c = db.query(...);
while (c.moveToNext()) {
    String name = c.getString(0);
    ...
}
```

```
SQLiteDatabase db = helper.getWritableDatabase();
ContentValues cv = new ContentValues();
cv.put("column", value);
...
db.insertOrThrow("table", null, cv);
```

Read Getting started with android-sqlite online: https://riptutorial.com/android-sqlite/topic/4630/getting-started-with-android-sqlite

Chapter 2: SQLiteOpenHelper

Examples

SQLiteOpenHelper with fully qualified DB path name/databse in public folder

Normally Android-SQLiteOpenHelper does not allow fully qualified path names where the database should be stored. So public database files are not possible.

You can use the SQLiteOpenHelper with a custom path if you provide a custom ContextClass and if you have write access in the target directory.

```
public class DatabaseHelper extends SQLiteOpenHelper {
    private static final int DATABASE_VERSION = 3;
    .....

    DatabaseHelper(final Context context, String databaseName)
    {
       super(new DatabaseContext(context), databaseName, null, DATABASE_VERSION);
    }
}
```

And here is the custom DatabaseContext class that does all the magic

```
class DatabaseContext extends ContextWrapper {

    private static final String DEBUG_CONTEXT = "DatabaseContext";

    public DatabaseContext(Context base) {
        super(base);
    }

    @Override
    public File getDatabasePath(String name)
    {
        File sdcard = Environment.getExternalStorageDirectory();
        String dbfile = sdcard.getAbsolutePath() + File.separator+ "databases" +
File.separator + name;
        if (!dbfile.endsWith(".db"))
        {
            dbfile += ".db" ;
        }

        File result = new File(dbfile);

        if (!result.getParentFile().exists())
        {
            result.getParentFile().mkdirs();
        }

        if (Log.isLoggable(DEBUG_CONTEXT, Log.WARN))
        {
            Log.w(DEBUG_CONTEXT,
                    "getDatabasePath(" + name + ") = " + result.getAbsolutePath());
        }

        return result;
    }

    /* this version is called for android devices >= api-11. thank to @damccull for fixing
this. */
    @Override
    public SQLiteDatabase openOrCreateDatabase(String name, int mode,
SQLiteDatabase.CursorFactory factory, DatabaseErrorHandler errorHandler) {
        return openOrCreateDatabase(name,mode, factory);
    }

    /* this version is called for android devices < api-11 */
    @Override
    public SQLiteDatabase openOrCreateDatabase(String name, int mode,
SQLiteDatabase.CursorFactory factory)
    {
        SQLiteDatabase result = SQLiteDatabase.openOrCreateDatabase(getDatabasePath(name),
null);
        // SQLiteDatabase result = super.openOrCreateDatabase(name, mode, factory);
        if (Log.isLoggable(DEBUG_CONTEXT, Log.WARN))
        {
            Log.w(DEBUG_CONTEXT,
                    "openOrCreateDatabase(" + name + ",,) = " + result.getPath());
        }
        return result;
    }
}
```

This is a copy of my answer to SQLiteOpenHelper problem with fully qualified DB path name

Read SQLiteOpenHelper online: https://riptutorial.com/android-sqlite/topic/4650/sqliteopenhelper

Credits

S. No	Chapters	Contributors
1	Getting started with android-sqlite	CL., Community
2	SQLiteOpenHelper	k3b