Importing sqlite database into android

The article presents a class to use the existing S'Lite database. The class you're in allows you to update your database with DB_VERSION. By adjusting the S'Lite database file to create DB S'Lite, I recommend using DB Browser for S'Lite. Download and install. Create a database as a file, save it. Add a database file to the asset folder. This class allows you not only to copy an existing database, but also to update it according to the database number. package org.harrix.sqliteexample; import android.content.Context; import android.database.S'LException; import android.database.sqlite.S'LiteDatabase; import android.database.sqlite.S'LiteOpenHelper; Import java.io.File Import java.io.FileOutputStream import java.io.IOException Import java.io.InputStream import java.io.OutputStream; DatabaseHelper expands S'LiteOpenHelper - private static string DB_NAME - info.db; Private static string DB_PATH - ; Private static final int DB_VERSION No 1; private base mDataBase; mContext's private final context private boolean mNeedUpdate - false; Public databaseHelper (contextual context) - super (context, DB_NAME, null, DB_VERSION); If (android.os.Build.VERSION.SDK_INT - 17) DB_PATH - context.getApplicationInfo ().dataDir /databases/; else DB_PATH - /data/data/ - context.getPackageName () - /databases/; this.mContext - context; copyDataBase (); this.getReadableDatabase (); - Public invalid updateDataBase () throws IOException if (mNeedUpdate) - File dbFile - new file (DB_PATH and DB_NAME); if (dbFile.exists)) dbFile.delete() copyDataBase (); mNeedUpdate - false; - private Galilee checkDataBase () - File dbFile - new file (DB_PATH - DB_NAME); Return dbFile.exist (); - private void copyDataBase () - if (!checkDataBase()- this.getReadableDatabase (); Catch (IOException mIOException) - throw a new bug DB_NAME (ErrorCopyingDataBase); OutputStream mOutput - the new FileOutputStream (DB_PATH and DB_NAME); byte' mBuffer - new byte (1024); int mLength; While ((mLength and mInput.read (mBuffer)) zgt; 0) mOutput.write (mBuffer, 0, mLength); mOutput.flush(); mOutput.close(); mInput.close(); - public boolean openDataBase () throws S'LException - mDataBase - S'LiteDatabase.openDatabase (DB_PATH - DB_NAME, null, SQLiteDatabase.CREATE_IF_NECESSARY); return mDataBase ! @Override public synchronized void is close - if (mDataBase !) null) mDataBase.close(); super.close (); - @Override public void onCreate (S'LiteDatabase db) - @Override public void on Upgrade (S'LiteDatabase db, int oldVersion, int newVersion) Package android.content.Context;import android.database.S'LException;import android.database.sqlite.s'LiteDatabase;import android.database.sqlite.S'LiteOpenHelper;import.io.FileOutputStream; import java.io.IOException; java.io.input DB_NAME Stream Private static string DB_PATH and; Private static final int DB_VERSION No 1; private base mDataBase; mContext's private final context private boolean mNeedUpdate - false; Public databaseHelper (context context) - super (context, DB_NAME, null, DB_VERSION); if (android.os.Build.VERSION.SDK_INT zgt; No. 17) DB_PATH context.getApplicationInfo ().dataDir/databases/; DB_PATH /data/data/ - context.getPackageName () - /databases/; this.getReadableDatabase(); public invalid updateDataBase () throws IOException - File dbFile - new file (DB_PATH and DB_NAME); private Galilee checkDataBase () - File dbFile - new file (DB_PATH DB_NAME); private void copyDataBase () - this.getReadableDatabase (); Catch (IOException mIOException) - throw a new bug (ErrorCopyingDataBase); private void copyDBFile () throws IOException - InputStream mInput - mContext.getAssets ().open (DB_NAME); InputStream mInput - mContext.getResources ().openRawResource (R.raw.info); OutputStream mOutput - the new FileOutputStream (DB_PATH and DB_NAME); byte' mBuffer - new byte (1024); While ((mLength and mInput.read (mBuffer)) zgt; 0) mOutput.write (mBuffer, 0, mLength); public boolean openDataBase () throws S'LException - mDataBase - S'LiteDatabase.openDatabase (DB_PATH DB_NAME, null, SQLiteDatabase.CREATE_IF_NECESSARY); return mDataBase ! Public synchronized void is close () - public void onCreate (S'LiteDatabase db) - public void on Upgrade (S'LiteDatabase db, int oldVersion, int newVersion) - if (new version of the zgt; oldVersion) Use of class in activity class, declare variables. In the onCreate method, write the following code. mDBHelper - a new databaseHelper (it); try mDBHelper.updateDataBase Catch (IOException mIOException) - throw a new bug (ImpossibleToUpdateDatabase); mDBHelper.updateDataBase(); catch (IOException - throw a new bug (ImpossibleToUpdateDatabase); mDb and mDBHelper.getWritableDatabase catch (S'LException mS'LException) - Database file in res/raw If you add a database file to the res/raw folder, use the following class modification. package org.harrix.sqliteexample; import android.content.Context; import android.database.S'LException; import android.s.database.sqlite. import android.database.sqlite.S'LiteOpenHelper; Java.io.File imports Java.io.FileOutputStream imports; Java.io.IOException imports; Java.io.InputStream imports Java.io.OutputStream imports DatabaseHelper expands S'LiteOpenHelper - a private static line DB_PATH DB_NAME private static final int DB_VERSION No 1; private base mDataBase; mContext's private final context private boolean mNeedUpdate - false; Public databaseHelper (Context Context) - super (context, DB_NAME, null, DB_VERSION); if (android.os.Build.VERSION.SDK_INT zgt;x 17) DB_PATH - context.getApplicationInfo ().dataDir/databases/; more DB_PATH - /data/data/ - context.getPackageName () - /databases/; this.mContext - context;

copyDataBase (); this.getReadableDatabase (); - Public invalid updateDataBase () throws IOException if (mNeedUpdate) - File dbFile - new file (DB_PATH and DB_NAME); if (dbFile.exists)) dbFile.delete.); copyDataBase (); mNeedUpdate - false; - private Galilee checkDataBase () - File dbFile - new file (DB_PATH - DB_NAME); dbFile.exist - private void copyDataBase () - if (!checkDataBase.)) - this.getReadableDatabase(); this.close (); Try copyDBFile ()DB_NAME - Catch (IOException mIOException) - Drop a new bug (ErrorCopyingDataBase); InputStream mInput - mContext.getResources ().openRawResource (R.raw.info); OutputStream mOutput - the new FileOutputStream (DB_PATH and DB_NAME); byte' mBuffer - new byte (1024); int mLength; While ((mLength and mInput.read (mBuffer)) zgt; 0) mOutput.write (mBuffer, 0, mLength); mOutput.flush(); mOutput.close(); mInput.close(); - public boolean openDataBase () throws S'LException - mDataBase - S'LiteDatabase.openDatabase (DB_PATH - DB_NAME, null, SQLiteDatabase.CREATE_IF_NECESSARY); return mDataBase ! @Override public synchronized void is close - if (mDataBase !) null) mDataBase.close(); super.close (); - @Override public emptiness onCreate (S'LiteDatabase db) - @Override public void on Upgrade (S'LiteDatabase db, int oldVersion, int newVersion) package org.harrix.sqliteexample;import android.content.Context;import android.database.s'LException;import android.database.sqlite.s'1 LiteDatabase;import android.database.sqlite.S'LiteOpenHelper;import java.io.FileOutputStream;import java.io.IOException;import java.ioException;import java.ioException;import java.io.OutputStream; public-class databaseHelper expands S'LiteOpenHelper - private static DB_NAME and info.db; Private static string DB_PATH and; Private static final int DB_VERSION No 1; private base mDataBase; mContext's private final context private boolean mNeedUpdate - false; Public databaseHelper (context context) - super (context, DB_NAME, null, DB_VERSION); if (android.os.Build.VERSION.SDK_INT zgt; No. 17) DB_PATH context.getApplicationInfo ().dataDir/databases/; DB_PATH /data/data/ - context.getPackageName () - /databases/; this.getReadableDatabase(); public invalid updateDataBase () throws IOException - File dbFile - new file (DB_PATH and DB_NAME); private Galilee checkDataBase () - File dbFile - new file (DB_PATH DB_NAME); private void copyDataBase () - this.getReadableDatabase (); Catch (IOException mIOException) - throw a new bug (ErrorCopyingDataBase); private void copyDBFile () throws IOException /InputStream mInput - mContext.getAssets ().open (DB_NAME); InputStream mInput - mContext.getResources ().openRawResource (R.raw.info); OutputStream mOutput - the new FileOutputStream (DB_PATH and DB_NAME); byte' mBuffer - new byte (1024); While ((mLength and mInput.read (mBuffer)) zgt; 0) mOutput.write (mBuffer, 0, mLength); public boolean openDataBase () throws S'LException - mDataBase - S'LiteDatabase.openDatabase (DB_PATH DB_NAME, null, SQLiteDatabase.CREATE_IF_NECESSARY); return mDataBase ! public synchronized emptiness close () - public void onCreate (S'LiteDatabase db) - public void on Upgrade (S'LiteDatabase db, int oldVersion, int newVersion) - if (newVersion zgt; oldVersion) oldVersion) oldVersion)