Библиотека Cupboard

Библиотека **Cupboard** создана для быстрого и удобного создания базы данных на основе класса-модели.

Домашняя страница библиотеки - <https://bitbucket.org/qbusict/cupboard>

Создавать базу данных с таблицей достаточно муторное дело с большим количеством кода. Библиотека **Cupboard** позволяет немного автоматизировать процесс. Вы создаёте класс с нужными полями и библиотека самостоятельно генерирует и обновляет таблицы, используя поля класса.

Рассмотрим простейший вариант для понимания.

Прописываем зависимость в **build.gradle**:

compile 'nl.qbusict:cupboard:2.1.2'

Создадим класс-модель с тремя полями. Каждое поле станет именем столбца в таблице.

package ru.alexanderklimov.testapplication;

public class Cat {

public Long \_id;

public String name;

public int age;

public Cat() {

this.name = "noName";

this.age = 0;

}

public Cat(String name, int age) {

this.name = name;

this.age = age;

}

@Override

public String toString() {

return "Cat{" +

"\_id=" + \_id +

", name='" + name + '\'' +

", age=" + age +

'}';

}

}

Здесь нужно обратить внимание, что поле **\_id** должно быть **Long**, а не **long**. Это связано с тем, что примитивный тип **long** имеет значение по умолчанию, и при его использовании все новые данные будут затирать предыдущие данные и ваша таблица будет содержать только один ряд.

Создадим класс на основе **SQLiteOpenHelper**, как обычно мы делаем при работе с SQLite. Но много кода писать не будем, а просто зарегистрируем созданную модель.

package ru.alexanderklimov.testapplication;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import static nl.qbusict.cupboard.CupboardFactory.cupboard;

public class CupBoardDatabaseHelper extends SQLiteOpenHelper {

private static final String DATABASE\_NAME = "cupboard.db";

private static final int DATABASE\_VERSION = 1;

public CupBoardDatabaseHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

static {

// Регистрируем модель

cupboard().register(Cat.class);

}

@Override

public void onCreate(SQLiteDatabase db) {

cupboard().withDatabase(db).createTables();

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

cupboard().withDatabase(db).upgradeTables();

}

}

Создадим активность из кнопки и текстовой метки. При запуске запишем данные, а при щелчке прочитаем первый элемент таблицы.

package ru.alexanderklimov.testapplication;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.support.v7.app.ActionBarActivity;

import android.view.View;

import android.widget.TextView;

import static nl.qbusict.cupboard.CupboardFactory.cupboard;

public class MainActivity extends ActionBarActivity {

private SQLiteDatabase mDB;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

setTitle("SQLite");

CupBoardDatabaseHelper dbHelper = new CupBoardDatabaseHelper(this);

mDB = dbHelper.getWritableDatabase();

Cat barsik = new Cat("Васька", 9);

cupboard().withDatabase(mDB).put(barsik);

Cat ryzhik = new Cat("Рыжик", 11);

cupboard().withDatabase(mDB).put(ryzhik);

}

public void onClick(View view) {

TextView infoTextView = (TextView) findViewById(R.id.textViewInfo);

// получим первый элемент таблицы

Cat firstCat = cupboard().withDatabase(mDB).query(Cat.class).get();

infoTextView.setText(firstCat.toString());

}

}

Получим строку таблицы по идентификатору:

// По \_id

Cat catID = cupboard().withDatabase(mDB).get(Cat.class, 6L);

infoTextView.setText(catID.toString());

Получим имя таблицы:

String table = cupboard().withEntity(Cat.class).getTable();

// Укороченный вариант

String table = cupboard().getTable(Cat.class);

infoTextView.setText(table);

Получим список записей:

Cursor cursor = cupboard().withDatabase(mDB).query(Cat.class).getCursor();

Iterable<Cat> itr = cupboard().withCursor(cursor).iterate(Cat.class);

String result = "";

for (Cat cat : itr) {

result = result + cat.toString() + "\n";

}

infoTextView.setText(result);

Получим результат по имени

// Получим первый подходящий элемент по имени

Cat cat = cupboard().withDatabase(mDB).query(Cat.class).withSelection( "name = ?", "Рыжик").get();

infoTextView.setText(cat.toString());

Остальные примеры смотрите на странице библиотеки.