先说明一下这个项目不允许用框架，但是以后在开发的过程中可以根据情况使用来加速进程。

LitePal项目主页

<https://github.com/LitePalFramework/LitePal>

郭霖专栏

<http://blog.csdn.net/guolin_blog/article/details/40083685>

开源的Android数据库框架，采用对象关系映射（ORM）的模式

(面向对象的语言和面向关系的数据库之间建立一种映射关系)

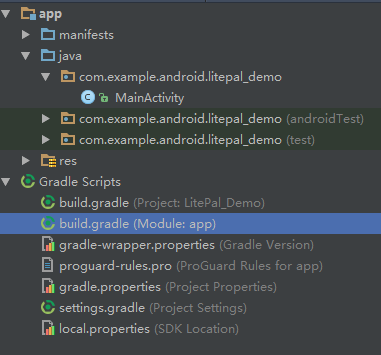
(面向对象的思维操作数据库！)

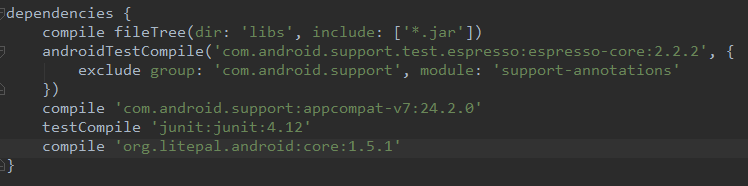
常用功能进行封装，不用编写SQL语句即可完成各种建表和增删查改的操作

编辑app/build.gradle

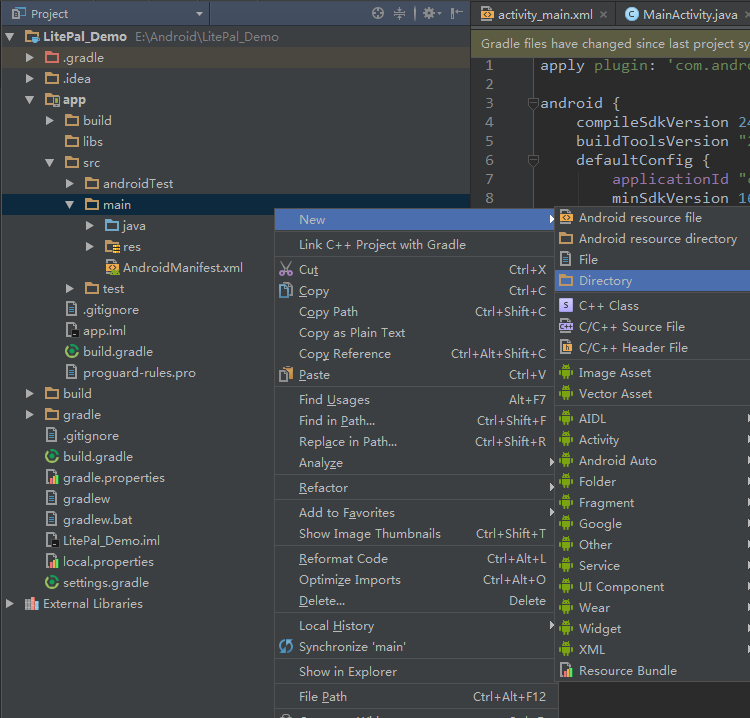
在dependencies闭包中添加

compile 'org.litepal.android:core:1.5.1'(填写最新版本号)



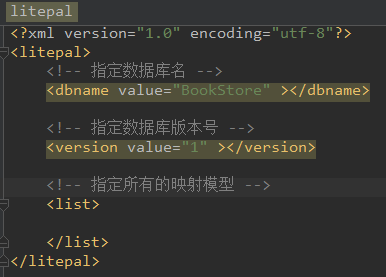


然后新建assets目录，再在assets下新建litepal.xml

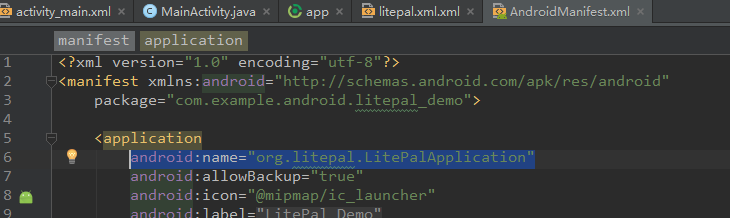




添加内容



修改AndroidMainfest.xml

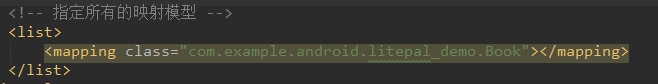


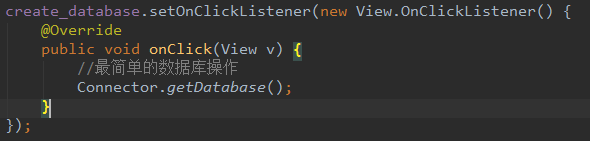
配置工作完毕。

新建Book类

Book类对应数据库中的Book表，而类中的每一个字段分别对应了表中的每一个列，对象关系映射最直观的体验。

Book添加到映射模型列表当中



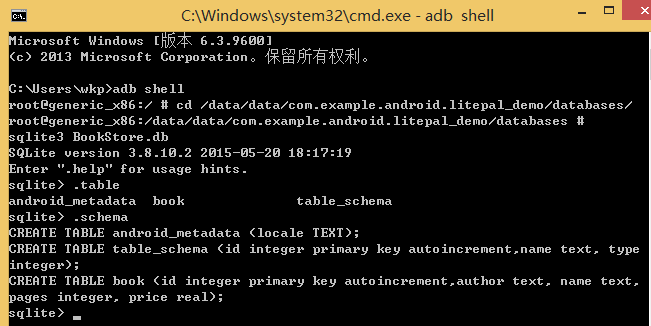


点击后查看创建状态

Android\_metadata 忽视

table\_schema表示LitePal内部使用的

Book自动生成

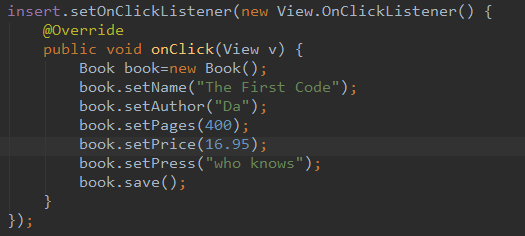


使用LitePal升级数据库简单

直接修改类，版本号+1，还会自动保留之前数据

实体类继承DataSupport

增加



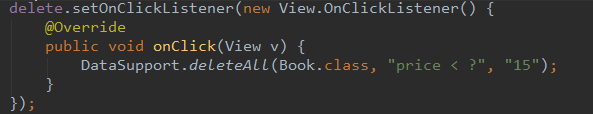
更新：对已存储的对象调用save方法

两种情况下 model.isSaved()方法返回true

1. 已经调用过model.save()方法去添加数据
2. Model对象是从数据库中查到的对象



删除：



查询：

findAll()方法返回值是一个Book类型的List集合，不用像之前那样再通过Cursor对象一行行去取值，LitePal帮助我们完成了赋值操作。

DataSupport.findBySQL()返回Cursor对象

