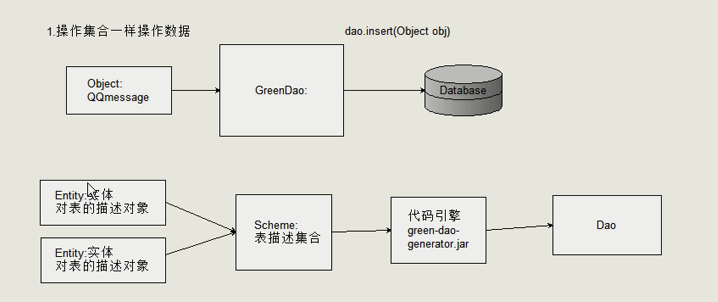
详细讲解：

http://www.open-open.com/lib/view/open1438065400878.html

原理：

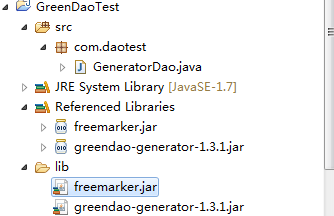


1. 将要保存到数据库的java对象描述成一个Entity实体对象
2. 将描述好的Entity实体对象保存到Scheme集合中，该集合用于保存所有的Entity实体描述对象
3. 通过GreenDao中的代码引擎，生成Entity在数据库对应的表的增删改查操作的Dao类

**操作步骤：**

**一、创建一个java工程，在java工程中通过greenDao的API创建表的实体Entity和schema，然后生成实体的dao和类到另一个Android工程目录下（通过在代码中指定路径）**

1. 创建一个Java工程，将freemarker.jar和greendao-generator-1.3.1.jar两个jar包拷贝到lib目录下，然后build to path



1. 指定类的Entity实体，通过greenDao提供的API生成实体的dao数据库操作类和实体类
2. 创建表的描述集合Schema用于保存实体对象

Schema schema = **new** Schema(数据库版本, 生成的实体类的dao保存路径) ;

Schema schema = **new** Schema(1,"com.example.im.dao") ;

1. 创建表的实体对象，指定对象的属性，属性名则会被作为字段名保存在数据库中 实体类类名

Entity entity = schema.addEntity("Contact");

//long型的属性 account

entity.addLongProperty("account");

//String类型的属性 nick

entity.addStringProperty("nick");

//int类型的属性 avator

entity.addIntProperty("avator");

//为该实体对应的数据库中的表添加一个自增长的主键

entity.addIdProperty().primaryKey().autoincrement() ;

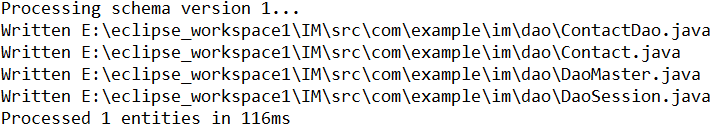
1. 创建出实体类和该实体类对应的dao

**new** DaoGenerator().generateAll(schema, "../IM/src");

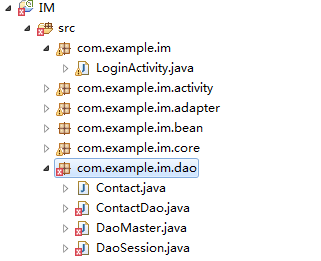
实体类和dao保存的路径，一般指定到目标Android工程中

../IM/src表示回到当前工作区间的IM安卓工程下的src目录下的com.example.im.dao（第一步中指定好的）目录中，将生成的实体类和其dao保存到该路径下

控制台输出执行成功的结果：



在IM/src/com.example.im.dao目录下生成了四个类：

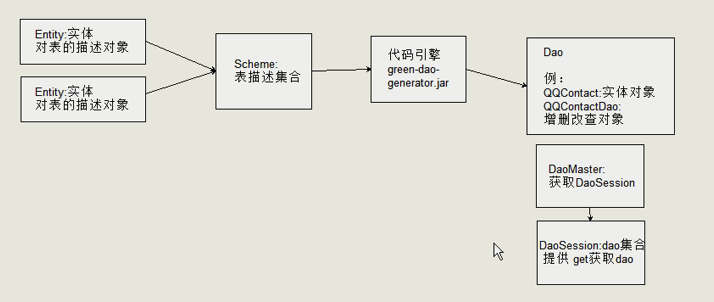


生成的四个类报错，原因是该Android工程下调用这四个类需要greenDao提供的API，解决方法：

将greendao-1.3.0-beta-1.jar包复制到IM安卓工程的libs目录下，然后build，问题解决

注：整个过程需要用到3个jar包

在目标Android工程中，通过调用上述步骤生成的实体对象和实体对象的dao，在数据库中对实体进行增删改查操作



当有多个实体时，会创建多个Dao，所有的Dao都保存在DaoSession中，DaoMaster则提供了所有Dao在DaoSession中的目录

**get()**

**get()**

DaoSession

get()

DaoMaster

1. 创建数据库

// 创建一个数据库

SQLiteDatabase db = **new** DaoMaster.DevOpenHelper(mContext, "im.db", **null**).getWritableDatabase() ;

1. 获取对数据库进行操作的dao（通过DaoMaster获取到DaoSession，再通过DaoSession获取到实体类的dao）

DaoMaster master = **new** DaoMaster(db);

DaoSession daoSession = master.newSession();

ContactDao contactDao = daoSession.getContactDao();

1. 创建一个示例实体对象，用于将其增删查改到数据库

// 创建一个实体对象

Contact contact = **new** Contact();

contact.setAccount(101L);

contact.setAvator(R.drawable.***ic\_launcher***);

contact.setNick("Michael");

1. 将实体对象插入到数据库中

// 将实体对象保存到数据库中

contactDao.insert(contact);

1. 在数据库中查询实体对象
2. 单条记录查询

Contact queriedItem = contactDao.load(1L);

查询结果：



1. 查询出所有的记录（保存的实例对象），查询结果保存在一个List集合中

List<Contact> contactList = contactDao.loadAll();

查询结果：



1. 更新上面查询到的实体对象的数据

queriedItem.setNick("Kobe Bryant");

contactDao.update(queriedItem);



1. 删除

1）直接删除某个实体记录

contactDao.delete(entityObject);

2）通过查找id删除数据库中保存的实体记录

contactDao.deleteByKey(2L);