**PKLormapping 1.1 框架介绍**

1. **特性**

* **支持面向对象，零SQL操作数据库**
* **支持模拟器版本和真机版本**
* **支持自定义SQL扩展**
* **支持1级缓存，万级数据可达到百倍速度获取**
* **支持c中基础类型映射，推荐使用NSNumber进行封装**
* **支持分页查询，排序**
* **支持线程池，默认为1，可自定义**
* **支持多线程并发处理，线程安全**
* **支持HQL语句**
* **目前不支持join连接方式自动补全**

**1.1版本**

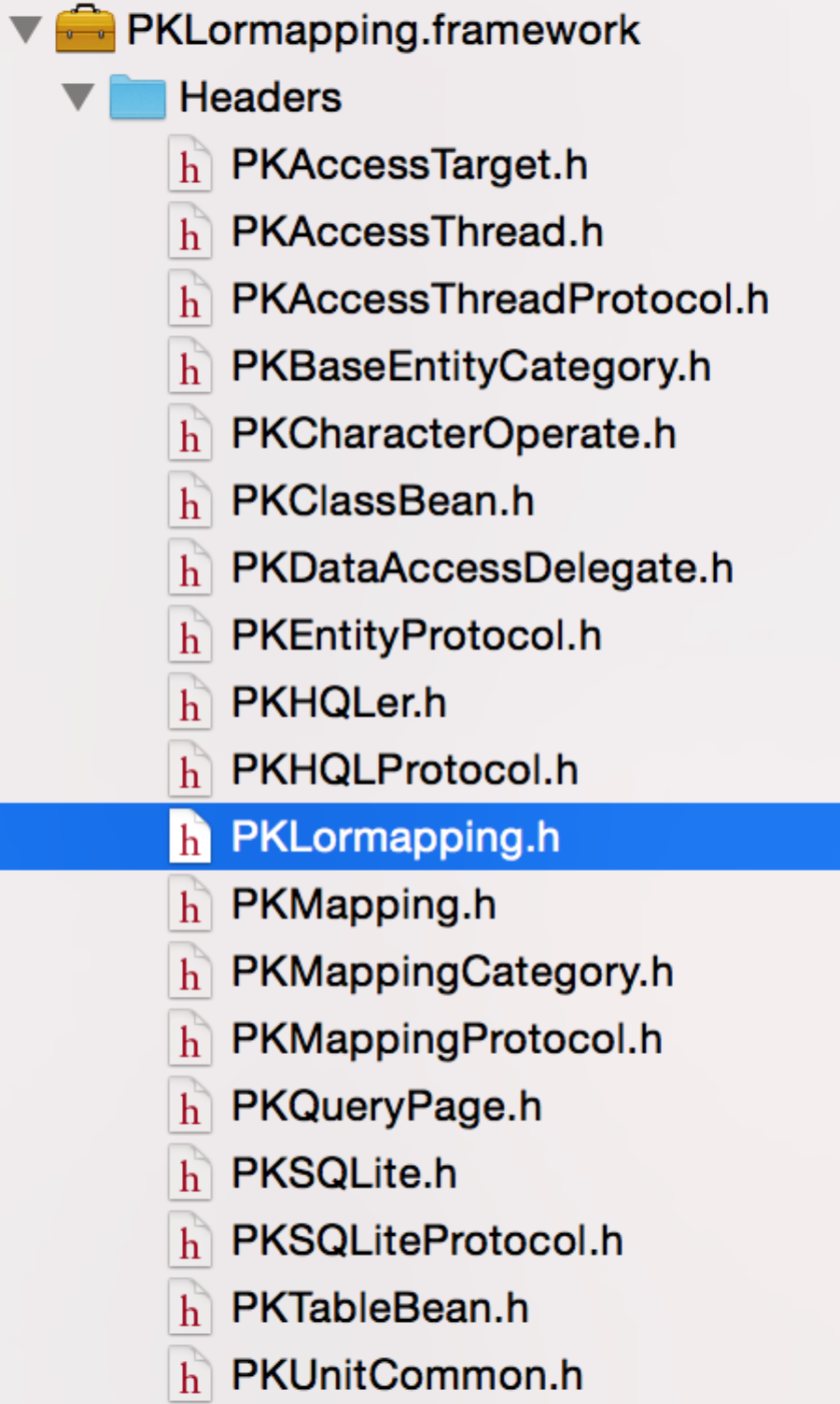
**修复：**

* **属性空值SQL映射失败BUG**

**新增：**

* **新增多表联动映射**
* **新增事务处理机制**
* **新增批量处理API，实测万级数据插入快3倍以上**

1. **接口说明**

****

**开发中常用接口(可点开框架接口查看)：**

1. PKLormapping ：框架头文件
2. PKHQLer ：HQL帮助类，负责组装条件查询，分页和排序
3. PKQueryPage ：分页类，负责分页
4. PKAccessThread ：数据库访问线程

映射表命名规则

表名 ：T\_XXX\_XX

字段名 ：XXX\_XX

命名中数字不用下横线分割

映射对象命名规则

类名：驼峰命名方式，每个峰映射表名中的下横线，如：UserInfo -> T\_USER\_INFO

属性名：驼峰命名方式，每个峰映射字段中的下横线，如：userId -> USER\_ID

类型：int,float,double类型最好使用NSNumber 存储，否则无法识别是否为空 命名中数字不用下横线分割

1. PKDataAccessDelegate ：数据回调接口
2. PKBaseEntityCategory ： 基础类类别
3. **Demo**

//创建数据库访问线程池

PKAccessThread ＊thread = [[PKAccessThread alloc] initWithSQLitePath:@"lorm.sqlite"];

thread.poolCount = 2;//线程池并发数，默认1

//创建某个对象

UserInfo \*userInfo = [[UserInfo alloc] init];

userInfo.name = @"周先生";

userInfo.sex = @"男";

userInfo.age = 101;

userInfo.number = [NSNumber numberWithDouble:123.334];

userInfo.date = [NSDate date];

//创建HQL帮助类

PKHQLer \*hql = [[PKHQLer alloc] initForEntity:userInfo];

[hql addLike:@"name" value:@"周"];//添加查询条件

//创建分页类

PKQueryPage \*queryPage = [[PKQueryPage alloc] init];

queryPage.rows = 100;//每页行数

queryPage.page = 0;//第x页

hql.queryPage = queryPage;//放入HQL帮助类

//创建访问线程加入线程池

[thread queryExecute:hql injectObj:userInfo callBackTarget:self];//根据HQL条件查询对象对应的表的数据，

[thread execute:sql callBackTarget:self];//自定义扩展SQL

[thread insertExecute:userInfo callBackTarget:self];//并发插入数据

[thread updateExecute:nil injectObj:userInfo callBackTarget:self];//更新数据

[thread countExecute:nil injectObj:userInfo callBackTarget:self];//计算总条数

[thread deleteExecute:nil injectObj:userInfo callBackTarget:self];

//结果集回调函数

//rs为返回的数据列表，state为SQL执行状态，YES为执行成功，NO为执行失败。

-(void) dataResult:(id)rs state:(BOOL)state{}

UserInfo \*userInfo = [[UserInfo alloc] init];//主表对象

PKMultipleEntityBean \*multipleEntiryBean = [[PKMultipleEntityBean alloc] init];//多表映射设置

multipleEntiryBean.mappingClass = [AddressBook class];//映射类

multipleEntiryBean.foreignKeyMapping = [NSDictionary dictionaryWithObject:@"userId" forKey:@"id"];//映射字段

mappingClass ：映射数组的对象class

foreignKeyMapping : 外键映射 ,key 为主键 （主表）, value 为外键 (从表)

如：{ id : userId } 对应数据库语句 -> id （主表）= userId （从表）

PKArray \*addressBookArray = [[PKArray alloc] init];//创建映射数组对象

addressBookArray.multipleEntityBean = multipleEntiryBean;//把关联表映射信息放入数组

userInfo.addressBook = addressBookArray;//从表映射设置完毕后放入需要映射的属性

PKHQLer \*hql = [[PKHQLer alloc] initForEntity:userInfo];

[thread queryExecute:hql injectObj:userInfo callBackTarget:self];

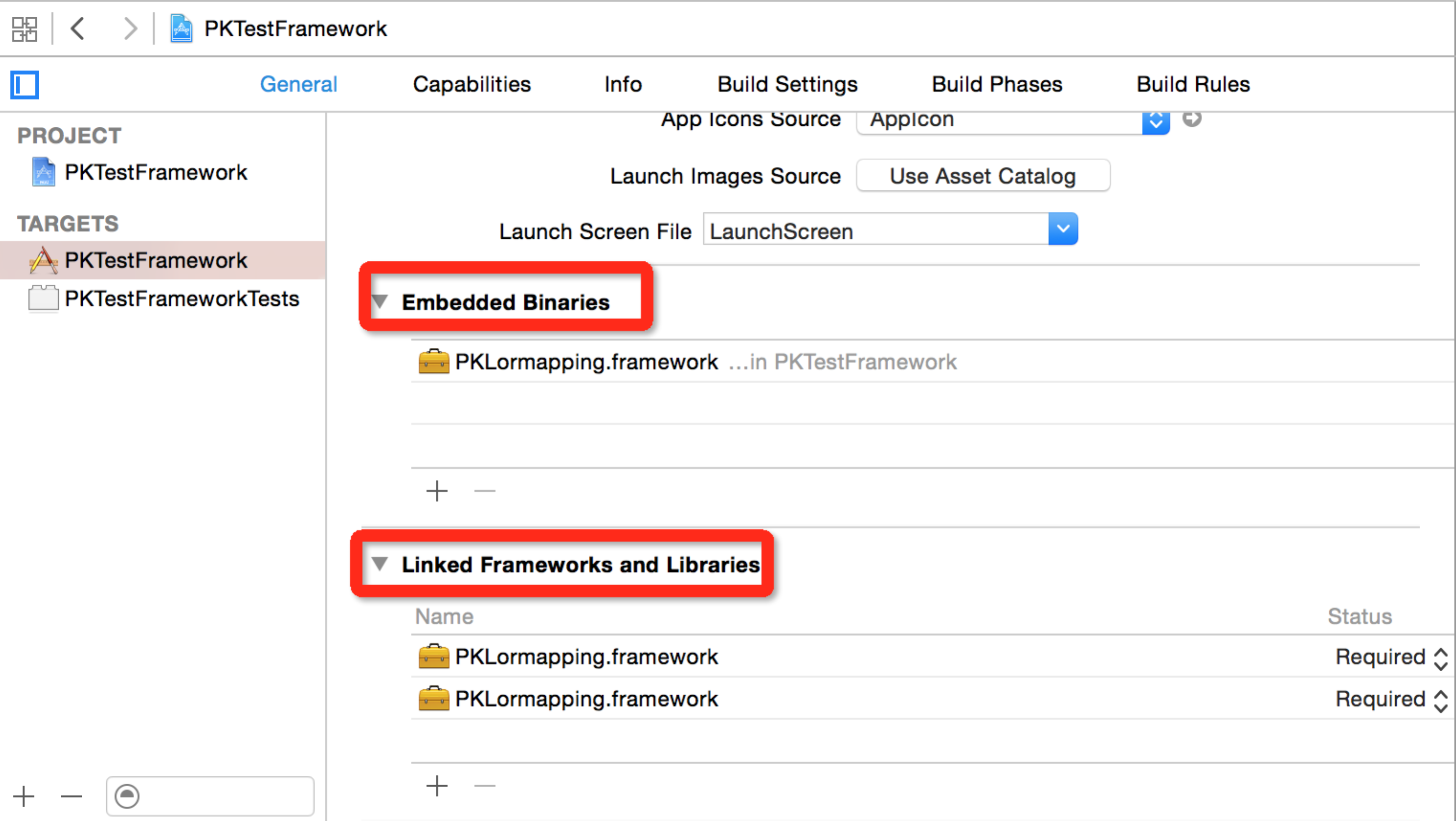
**PS：**

**主键和外键映射的属性非对象类型，必须用NSNumber类型才能自增长。**

**多表关联映射，映射数组必须为PKArray数组。**

**引用方式：#import "PKLormapping.framework/Headers/PKArray.h"**

1. **导入方式**

****

**头文件引用：**

#import "PKLormapping.framework/Headers/PKLormapping.h"

1. **反馈**

作者：周老师

QQ：974871365

邮箱：packyzhou@icloud.com

Blog: http://blog.csdn.net/packyzhou