

Wiley：移动开发工程师实验报告

云课堂昵称：MartinZhou

实验日期：2015.12.10

一、实验题目

在Xcode中创建和使用SQLite数据库

二、实验要求

- 1.在Xcode中创建并打开RegistrationForm应用
- 2.添加数据库库到项目
- 3.通过terminal创建一个数据库
- 4.在Xcode中添加代码来打开、创建和关闭RegistrationForm应用中的数据库

三、操作步骤

- 1.本实验将会使用到的环境
 - (1) terminal
 - (2) Xcode
- 2.操作步骤
 - (1) 在Xcode中打开MZRegistrationForm应用。
 - (2) 选择 Linked Frameworks and Libraries 部分,然后点+图标来添加新库 , 如图1。

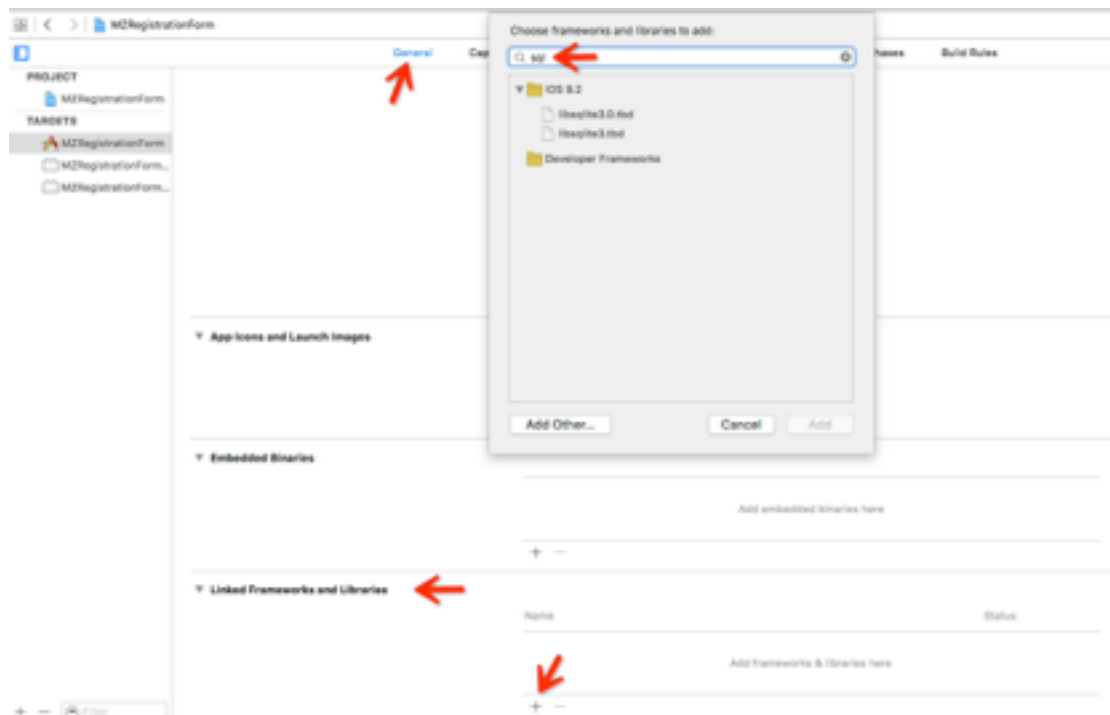
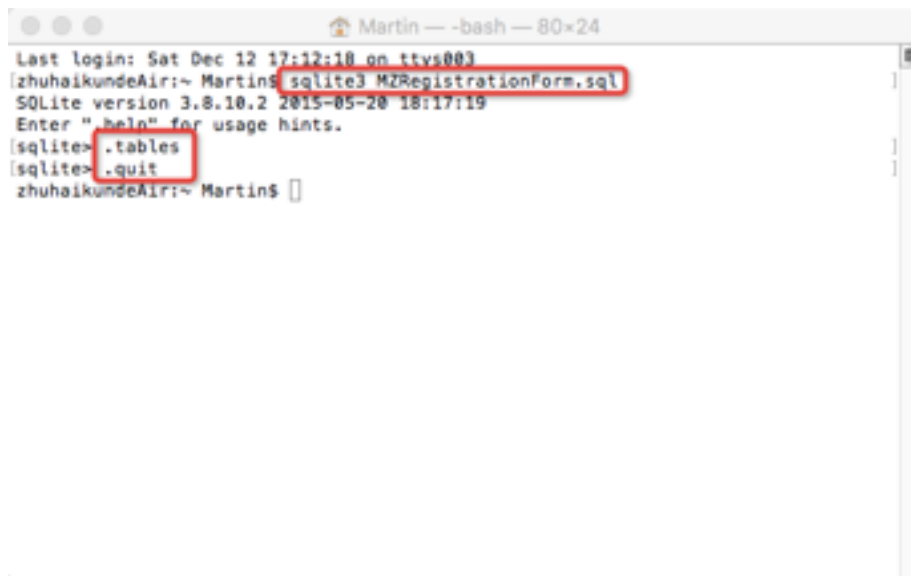


图 1

- (3) 在 Choose frameworks and libraries to add 对话框中,搜索 libsqlite3.tbd 文件

并点 Add 按钮。

(4) 之后使用 terminal 命令创建一个 SQLite 数据库。首先,输入# sqlite3 MZRegistrationForm.sql 并使用如下命令创建一张表。.table 命令被用于表的检查,.quit 命令被用于退出 SQLite 数据库,见图2。

A screenshot of a terminal window titled "Martin — -bash — 80x24". The terminal shows the following commands and output: "Last login: Sat Dec 12 17:12:10 on ttys003", "[zhuhaikundeAir:~ Martin\$ sqlite3 MZRegistrationForm.sql]", "SQLite version 3.8.10.2 2015-05-20 18:17:19", "Enter \"help\" for usage hints.", "[sqlite> .tables]", "[sqlite> .quit]", and "zhuhaikundeAir:~ Martin\$". The commands "sqlite3 MZRegistrationForm.sql", ".tables", and ".quit" are highlighted with red boxes.

```

Last login: Sat Dec 12 17:12:10 on ttys003
[zhuhaikundeAir:~ Martin$ sqlite3 MZRegistrationForm.sql
SQLite version 3.8.10.2 2015-05-20 18:17:19
Enter "help" for usage hints.
[sqlite> .tables
[sqlite> .quit
zhuhaikundeAir:~ Martin$
```

图 2

(5) 要添加 MZRegistrationForm.sql 到 Xcode,从 File 菜单选择 Add Files to “MZRegistrationForm”选项,见图3。

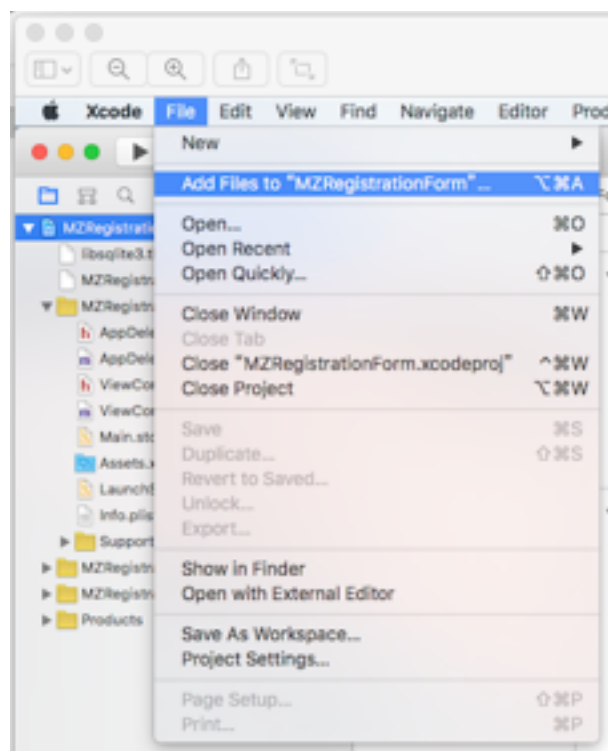


图 3

(6) 选择你所创建的 `MZRegistrationForm.sql`。选择 **Copy items into destination group's folders** 复选框,然后点 **Add**,如图 6 所示。这会添加 `MZRegistrationForm.sql`数据库文件

(7) 之后编辑 `ViewController.h` 文件,如下所示:

a. 获得 `databasePath` 及 `contactDB` 的代码:

```
- (IBAction)Submit:(id)sender;

// Save database path
@property (strong, nonatomic) NSString *databasePath;

// Contact database
@property (nonatomic) sqlite3 *contactDB;
```

b. 在 `ViewController.m` 文件中,像下面这样编辑 `(void)viewDidLoad` 方法:

写代码来辨别应用的文档目录并构造一个路径到 `MZRegistrationForm.db` 数据库文件。

写代码来创建一个 `NSFileManager` 实例,来探测数据库文件是否存在。如果数据库文件不存在,代码会将 路径转化为 UTF-8 字符串并使用 `SQLite sqlite3_open()` 函数创建一个数据库。

最后,写代码来关闭数据库。

```
- (void)viewDidLoad {
    [super viewDidLoad];
    NSString *docsDir;
    NSArray *dirPaths;

    // Get document Directory path
    dirPaths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
    NSUserDomainMask, YES);
    docsDir = dirPaths[0];

    // Build path to the database file
    _databasePath = [[NSString alloc]
    initWithString:[docsDir stringByAppendingPathComponent:@"MZRegistrationForm.sql"]];
    NSFileManager *filemgr = [NSFileManager defaultManager];
    if ([filemgr fileExistsAtPath: _databasePath] == NO)
    {
        const char *dbpath = [_databasePath UTF8String];
        if (sqlite3_open(dbpath, &_contactDB) == SQLITE_OK)
        {
            char *errMsg;

            // Adding database if not present
            const char *sql_stmt = "CREATE TABLE IF NOT EXISTS MZRegistrationForm
            (EnterName text, EnterEmail text, EnterPhone text, EnterCompany text, EnterId text)";
            if (sqlite3_exec(_contactDB, sql_stmt, NULL, NULL, &errMsg) != SQLITE_OK)
                {NSLog(@"Failed to create table");}

            //Closing database
            sqlite3_close(_contactDB);
        } else {
            NSLog(@"Failed to open/create database");
        }
    }
}
```

c. 在 `ViewController.m` 文件中,编辑 `(void)saveData:(id)sender` 方法,做法如下:

```
- (void)Submit:(id)sender {

    // Action when tap on background

    NSLog(@"Form Submitted");

    sqlite3_stmt *statement;
    const char *dbpath = [_databasePath UTF8String];
    if (sqlite3_open(dbpath, &_contactDB) == SQLITE_OK)
    {
        NSString *insertSQL = [NSString stringWithFormat:@"INSERT INTO
MZRegistrationFrom (Name text, Email text, Phone text, Company text, ID text)
VALUES ('%@','%@','%@','%@','%@')",_EnterName.text,
_EnterEmail.text,_EnterPhone.text,_EnterCompany.text,_EnterId.text];
        const char *insert_stmt = [insertSQL UTF8String];
        sqlite3_prepare_v2(_contactDB, insert_stmt,-1, &statement, NULL);

        NSLog(@"DB Open");

        if (sqlite3_step(statement) == SQLITE_DONE)
        {
            _EnterName.text = @"";
            _EnterEmail.text = @"";
            _EnterPhone.text = @"";
            _EnterCompany.text = @"";
            _EnterId.text = @"";
        }

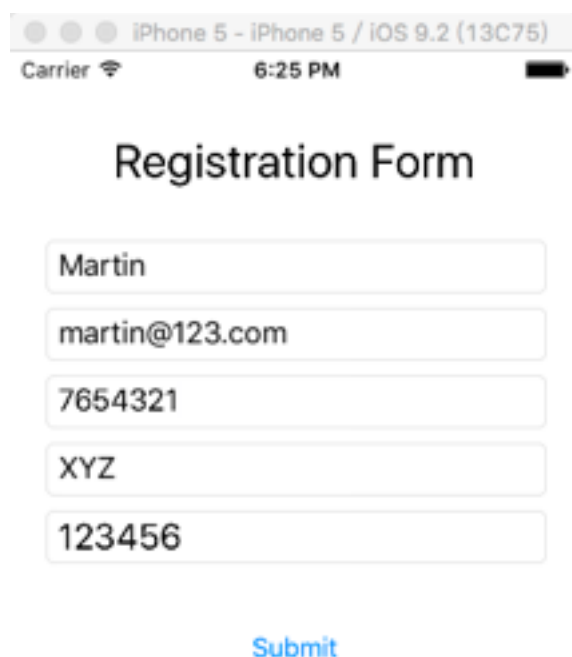
        sqlite3_finalize(statement);
        sqlite3_close(_contactDB);
    }
}
```

写代码来保存表单数据到数据库中。在 `saveData` 中进行添加。

这段代码会打开数据库文件,从表中提取文本,并将数据作为记录添加到数据库。

(8) 添加代码到 `ViewController.m` 文件后,运行程序。运行可以点 **Product** 菜单下的 **Run**,也可以点 **Run** 按钮。两者都能运行程序。

(9) 最后,注册表单得以创建,如图 4 所示。你可以添加细节到这个表单。



Registration Form

Martin

martin@123.com

7654321

XYZ

123456

Submit

四、实验结果

在模拟器中输入信息后,点击“Submit”,在控制台可以看到用来验证数据库被创建的NSLog内容被打印了出来。之后使用Terminal,进入到MZRegistrationForm.sql,通过.tables命令看表MZRegistrationForm是否被创建,并查看其中内容来验证数据已被存储。也可以使用导航栏的文件列表,直接点击MZRegistrationForm.sql文件查看其中的内容。

五、总结反思

- 1.回顾了UI的建立方法,可以熟练的控制不同控件的属性了。
- 2.了解了一些Terminal的使用方式,包括vim编辑器,sqlite3的简易使用方法及.tables、.exit等命令。
- 3.写代码的过程中,知道了一些比较重要的代码块:

(1) sqlite3_open(dbpath, &_contactDB)如果其值等于OK就表示数据库已成功打开。但是SQLite3是采用可移植的C编写的,它不知道什么是NSString,顾又有以下代码:

(2) `const char *dbpath = [_databasePath UTF8String];`
(3) 通过SQL的CREATE语句创建一个新表，并使用函数`sqlite3_exec`将其传递到打开的数据库。

```
char *errMsg;  
const char *sql_stmt = "CREATE TABLE IF NOT EXISTS MZRegistrationForm  
(EnterName text, EnterEmail text, EnterPhone text, EnterCompany text, EnterId text);  
if (sqlite3_exec(_contactDB, sql_stmt, NULL, NULL, &errMsg) != SQLITE_OK)  
{  
    NSLog(@"Failed to create table");  
}
```

六、作业提交

1. 请将本文档按照《Wiley移动开发_X章X节_云课堂昵称》的名称命名；
2. 请将本文档、源代码文件打包以附件形式上传到课程作业部分