RealmSwift入门

1、我使用的 pod 库，所以先 pod 库安装一下，安装完别忘了先编译一下，不然 import 不到头文件

```
platform :ios, '8.0'

target 'realmExmple' do
  use_frameworks!

pod 'RealmSwift'

end
```

2、在 AppDelegate 的 didFinishLaunchingWithOptions 中添加如下方法，引入头文件 import RealmSwift

```
 /* Realm 数据库配置，用于数据库的迭代更新 */
        let schemaVersion: UInt64 =
        // 数据库工具类的版本号
        SSRealmTool.schemaVersion = schemaVersion

                     let config = Realm.Configuration(schemaVersion: schemaVersion,
migrationBlock: { (migration, oldSchemaVersion) in
             /* 什么都不要做！Realm 会自行检测新增和需要移除的属性，然后自动更新
硬盘上的数据库架构 */
            if (oldSchemaVersion < schemaVersion) {
                print("----数据库升级")
            }
        })

        Realm.Configuration.defaultConfiguration = config
        Realm.asyncOpen { (realm, error) in
            /* Realm 成功打开，迁移已在后台线程中完成 */
            if let _ = realm {

                print("Realm 数据库配置成功")
            } else if let error = error { /* 处理打开 Realm 时所发生的错误 */

                print("Realm 数据库配置失败：\(error.localizedDescription)")
            }
        }
```

3、接下来开始写数据库工具类，新建一个 Swift 文件 SSRealmTool

```
class SSRealmTool {

    /// 数据库版本号
    static var schemaVersion: UInt64 =

    /// 唯一的数据库操作的 Realm
    static let ss_realm = realm()

    /// 获取数据库操作的 Realm
    private static func realm() -> Realm {

        // 获取数据库文件路径
        let fileURL = URL(string: NSHomeDirectory() + "/Documents/demo.realm")
        // 在 APPdelegate 中需要配置版本号时，这里也需要配置版本号
        let config = Realm.Configuration(fileURL: fileURL, schemaVersion: schemaVersion)

        return try! Realm(configuration: config)
    }

}
```

4、新建两个对象 SSDog SSPerson

```
class SSDog: Object {
    // 主键不自增，如果要主键的话，这样写不会有重复的主键
//    @objc dynamic var id = NSUUID().UUIDString
    // @objc dynamic **必须写
    @objc dynamic var name = ""
    @objc dynamic var age =
    @objc dynamic var date = Date()
    // 如果需要增加属性的话，只需要在 appdelegate 的版本号加 1 即可自动升级
//    @objc dynamic var weight = 1

    /// 所属人可选
    @objc dynamic var person: SSPerson?

    // 设置主键的方法
//    override static func primaryKey() -> String? {
//        return "id"
//    }
}

class SSPerson: Object {

    @objc dynamic var name = ""
    @objc dynamic var age =

    // 数组使用 List
    let dogs = List<SSDog>()
}
```

5、可以在 viewController 中调用我们的数据库了

5.1、添加一个 person

```
let realm = SSRealmTool.ss_realm

        let person = SSPerson()
        person.name = "person\(arc4random_uniform(1000))"
        person.age = Int(arc4random_uniform())

        for _ in ..< {

            let dog = SSDog()
            dog.name = "dog\(arc4random_uniform(1000))"
            dog.age = Int(arc4random_uniform())
            dog.person = person
            person.dogs.append(dog)
            print("添加一只狗：\(dog.name),age:\(dog.age),date:\(dog.date)")
        }

        try! realm.write {
            realm.add(person)
            print("添加一个人:\(person.name)，年龄:\(person.age)")
        }
```

5.2、查询

```
let realm = SSRealmTool.ss_realm
        print("总共有\(realm.objects(SSPerson.self).count)位人")
        print("总共有\(realm.objects(SSDog.self).count)只狗")

        if realm.objects(SSDog.self).count ==  || realm.objects(SSPerson.self).count ==  {
            return
        }
        let dogs = realm.objects(SSDog.self).filter("age <= 5")

        for dog in dogs {
            print("---查询狗:\(dog.name)，age:\(dog.age)，主人是：\(dog.person?.name ?? "无人")")
        }

        print("---------person---------")
let persons = realm.objects(SSPerson.self).filter("age <= 10")

        for person in persons {
            print("---查询name:\(person.name), 有\(person.dogs.count)只狗")
            print("-----每个人下面的狗-------")
            let someDogs = person.dogs.filter("age <= 5")
            for someDog in someDogs {
                    print("---查询狗:\(someDog.name), age:\(someDog.age)，主人是：\(someDog.person?.name ?? "无人")")
            }
        }
```

5.3、更新

```
let realm = SSRealmTool.ss_realm
        
        let dogs = realm.objects(SSDog.self).filter("age > 5")
        
        for dog in dogs {
            
            try! realm.write {
                
                print("---更新前name:\(dog.name), age:\(dog.age)")
                dog.name += "修改"
//                  realm.add(dog, update: true) // 这需要主键
            }
        }
        
        let dogs2 = realm.objects(SSDog.self).filter("age > 5")
        
        for dog in dogs2 {
            print("---更新后name:\(dog.name), age:\(dog.age)")
        }
```

5.4、删除

```
let realm = SSRealmTool.ss_realm
        
        let person = SSPerson()
        person.name = "person\(arc4random_uniform(1000))"
        person.age = Int(arc4random_uniform())
        
        for _ in ..< {
            
            let dog = SSDog()
            dog.name = "dog\(arc4random_uniform(1000))"
            dog.age = Int(arc4random_uniform())
            dog.person = person
            person.dogs.append(dog)
            print("添加一只狗：\(dog.name),age:\(dog.age),date:\(dog.date)")
        }
        
try! realm.write {
            realm.add(person)
            print("添加一个人:\(person.name)，年龄:\(person.age)")
}
```