Meet-iOS 开发相关文档

1. 整体结构

MVC，引入Pod管理一些开源库存，也有部分直接引入到FrameWork里

1. Model层
2. Dao文件对应单例对象，用于建表和对应的model的本地数据化操作（FMDB）。

父类为superDao，封装了一些对数据库操作

1. ModelBean 对应的Model文件 。用于封装网络数据和需要保存本地持久化的数据模型 。

父类为BaseModel ,实现了对模型的封装（-**initWithDictionary:**）方法。idKey :为数据库存主键，保存到数据库存自动添加

如需要保存到数据库，需要。1）把对应每个字段的键要暴露给相应的Dao如extern NSString \*const k\_User\_userId; (注意暴露的值不能相同 。2)必须实现**columnArray和valueArray。**且要相互对应的位置

3.UserInfo 单例(用户信息类)

|  |  |  |  |
| --- | --- | --- | --- |
| userId | 用户id，微信登陆后返回的id与WXUserInfo里的unionid 一致，用户维一标识 | 字符串 | 数据库存对应  k\_User\_userId  =@” User\_userId” |
| LoginTyep | 登录状态。0为退出状态。1为当前使用中 | NSNumber | k\_User\_loginType |
| name | 真实姓名 | NSString |  |
| city | 微信返回的 | NSString |  |
| country | 微信返回的 | NSString |  |
| headimgurl | 微信返回的，图像URL。修改后传入服务给的Url | NSString |  |
| sex | 性别(1为男，2为女，与微信给的保持一致) | NSNumber |  |
| **eMail** | 微信返回的.用户邮箱，提意见后保存 | NSString |  |
| modifySex | 是否修改过性别（0，1） | NSInteger |  |
| **brithday** | 生日 | NSString |  |
| **height** | 身高 | NSString |  |
| **phoneNo** |  | NSString |  |
| **WX\_No** | 微信号 | NSString |  |
| **workCity** | 工作生活城市 | NSString |  |
| **income** | 年收入 | NSString |  |
| **state** | 感情状态 | NSString |  |
| **home** | 家乡 | NSString |  |
| **constellation** | 星座 | NSString |  |

**4.MoreDescriptionModel 个人更多信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **UserId** | **同UserInfo里的userId** |  |  |
| **title** | **个人信息里的tilte** |  |  |
| **content** | **个人信息里描述内容** |  |  |
| **Index** | **此个人信息所在的顺序** |  |  |
| **HasImage** | **是否有图片** |  |  |
|  |  |  |  |

**5.** WXUserInfo类（单例微信用户登录成功后微信所返回的用户信息）

详见微信 API

<https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=open1419317853&lang=zh_CN>

得到后会写入UserInfo.Plist文件,存入的键为

NSString \*keyWXUserInfo = @"WeChatUserInfoKey";启动时会获取（有没有必要？）

**6.** WXAccessModel类（单例微信用户登录时获取到的token数据。微信所返回的信息）

详见微信 API

<https://open.weixin.qq.com/cgi-bin/showdocument?action=dir_list&t=resource/res_list&verify=1&id=open1419317853&lang=zh_CN>

得到后会写入UserInfo.Plist文件,存入的键为

static NSString \*keyAccessModelSave = @"accessModelSaveKey";启动时会获取（有没有必要？）

用于刷新token,和判断token,是否已经失效！

1. View层

主要是一些订制的Cell 和控件，不再赘述。

CoreTextView：第三方的，主要是用于对文字的排版。CoreTextEngine，里为唐巧写的coreText的教程…

1. Contrller层

MainViewControll 为Window下的rootController:包含Home，Mett Content,， Me

**Home**

分为两版，第二版类似网易新闻布局，由于担心早期数据少，所以暂时末用。

MeetBaseViewController：内部的NavigationContrller可继承之，实现了统一风格的Navigation订制，和popGestureReaginer返回方法，但要先设备delegate。

Mett Content,

我的订单（订单中心。）

BaseMeetViewController为些模块里的基类，实现了tableView创建和subContrllView的设置，下拉刷新等。

下面的继承于它的subContrllView。具体业务逻辑可在对应处实现

ME

我

Profile里为对应的填写个人信息，（MyProfileViewController）。界面上的显示逻辑较复杂化。

添加更多个人经历（AddInformationViewController）和编辑

描述个人亮点（AddStarViewController

个人更多信息的展示（MyDisplayViewController）,文字采用的CoerText排版，图片则还是用UIImageView加在scrollView上。

MyPhotosViewController（添加照片和拍照）保存图片。

保存的路径为Library ->Cache->UserInfo.userID为名的文件- >

1. headimg 为用户头像图片
2. MostContent 为更多用户信息里的图片，然后文件为添加图片时图片所在cell对应的indexPath.Section->IndexPath.row->Big or Small

Big 为浏览时的大图，small为显示在cell上的小图。对就的删除功能也在此内有实现

Login

登录

1, WeChatResgisterViewController

为微信的注册页面，包含新用户的邀请码，（待校检）。老用户直接登录（逻辑还没写）登录从后台拿到用户信息，查看本地是否存在。赋给UserInfo 然后保存（更新）到本地，保存要注意要重新从数据库取出mapping当前的UserInfo,因为当前的可能没有idKey，以后对其操作可能无法保存。

2 .WXLoginViewController

新用户注册类。注册后，将数据保存到数据库。需要mapping当前的UserInfo。获取idKey!!!

还有两个类是以后可能会用到的其它平台登录时的，暂时不必理会！

1. Global全局文件
2. AppData ,单例，全局的变量存放，和一些对文件操作等。
3. UITools，单例。常用的一些UI集中的处理，如订制Navigation操作。HUD的显示等
4. NetWorkObject ，对AFNetWorking, Http请求进行的封装，目前只有Get方法，downLoad，如有需要可再添加
5. StoreBoard

工程里大部分的Controller都包含对应的StoryBoard,。为了方便以后的维护，对应的功能模块的Board也被单独的拆分独立开来