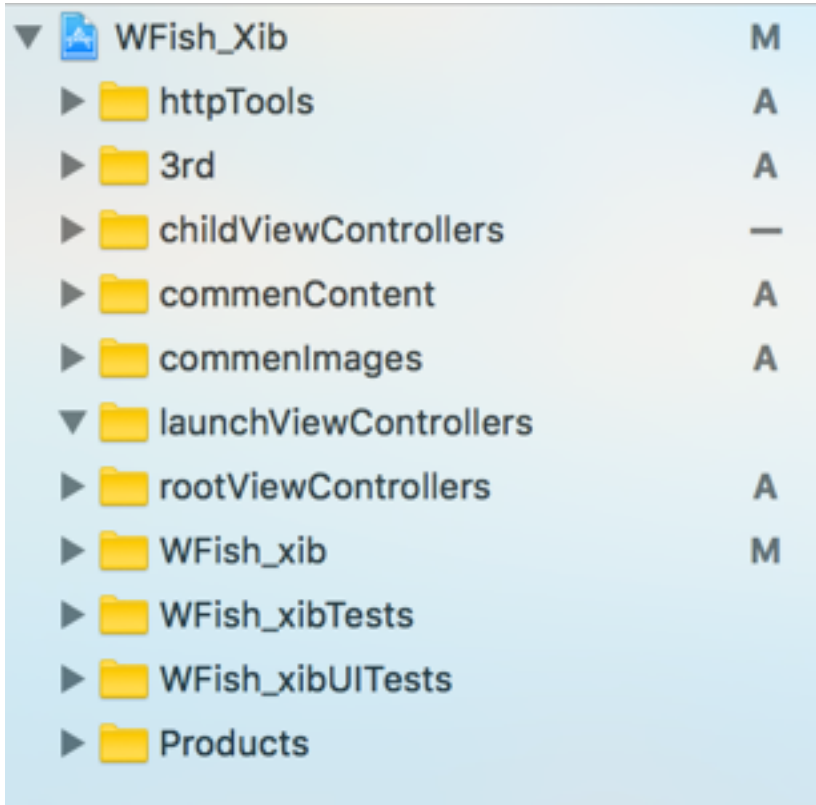


ZJITTeam多人合作框架(iOS端)使用说明

- 1.从gitHub上将MyApplication.zip文件克隆到自己的电脑(最好用sourceTree便于以后迭代)
- 2.解压后打开里面的demo文件(大致如图所示) 采取代码 + xib的方式进行开发

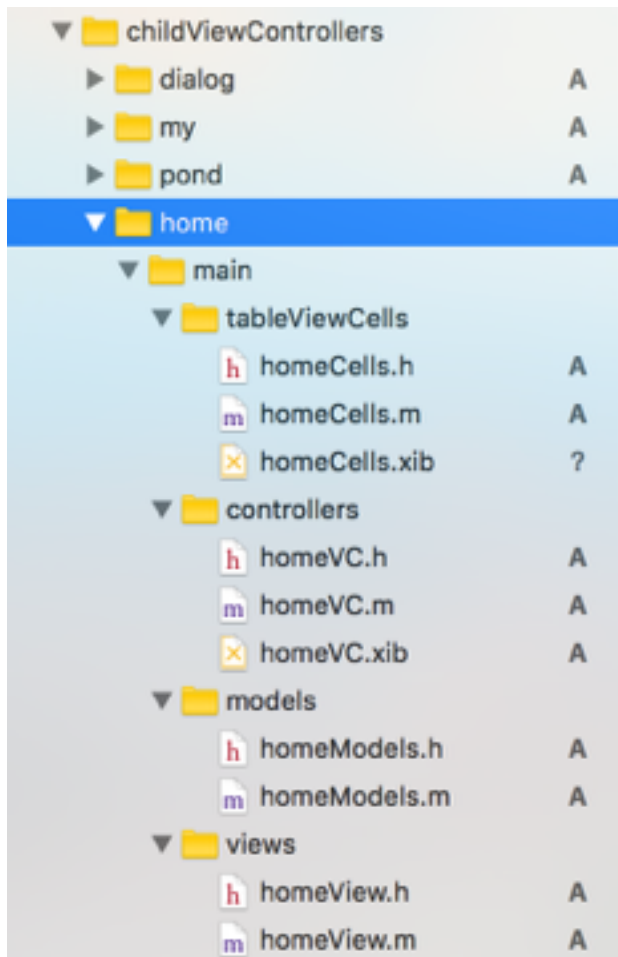


说明:

- 1.httpTools文件夹里面封装了网络请求，由我编写代码，你们不允许进行修改
- 2.3rd文件夹存放第三方框架，里面已经包含常用三方框架，你们也可以向里面添加其他第三方框架
- 3.childViewControllers存放项目主体部分，采用mvc模式，主体框架由我搭建，你们不允许新增文件夹，只可修改自己负责的部分文件
- 4.commonContent存放项目通用文件，你们没有编辑权限
- 5.launchViewControllers存放启动页面文件，由负责启动页面编码工作的同学负责
- 6.rootViewControllers存放根控制器文件，由我编码你们不允许修改
- 7.后面的配置文件由我负责，你们不允许修改

3.childViewControllers详细说明

文件夹细分为“tableViewCells”，“controllers”，“models”，“views”。其中tableViewCells文件夹存放xib cell文件以及继承自UITableViewCell的.h.m文件， controllers文件夹存放xib controller以及继承自UIViewController的.h.m文件， models文件夹存放继承自NSObject的.h.m文件， views文件夹存放继承自(UIView, UITableView, UIWebView等)文件。理论上models文件夹中内容由我编写提供不允许修改，其他文件夹文件由负责该模块的同学负责编写修改。



4.网络请求详细说明(详细看demo)

- 1.在需要进行网络请求的文件中引入commonHttpTools.h文件；
- 2.创建commonHttpTools实例，调用对应的网络请求方法
- 3.我已经将获取的数据转换成模型，通过block回调，你们只需在回调的block中获取对应数据模型即可
- 4.进行一些UI刷新的操作

5.关于版本迭代

统一使用sourceTree进行版本迭代，具体怎么使用可以自行百度

6.关于代码规范

属性命名统一采用驼峰规则如：

```
@property (strong, nonatomic) NSArray *pondNameArr;
```

方法命名尽量使之读起来通俗易懂如：

```
+ (instancetype)initHomeTitleCell:(UITableView *)tView andCellIdentify:(NSString *)cellIdentify;
```

不允许出现诸如l1, l2, l3等变量命名

7.后续规划

本框架暂为试行框架，以后可将网络请求这块封装成静态库