

页面编写习惯

1. 简介：每个人都有自己的页面编写习惯，下面就自己的编码习惯进行下介绍，后续，会不断进行更新，因为可能后续会学到更好的编码风格，目前主要以 mvvm 思想为基础进行编写：

2. 编码习惯 of cwn

(1).创建viewController.xib(为什么不是storyboard? 如果是storyboard的话svn或git管理开发难以避免冲突)，autolayout或者frame做好界面布局，建立outlet关联，列表设置delegate但不设置datasource为viewController

(2).创建viewControllerTableModel, 该类的基类充当列表数据源, 该类负责两个方面：

(1) 列表网络数据请求或者页面其它网络请求(比如请求广告轮播图片数组)，

(2) 其它业务逻辑处理(比如日期转换，判断某个时间是否在规定时间内)

(3).创建自定义cell，这个cell可以用xib创建(最便捷的方式)；当一个列表cell种类太多的时候，可以考虑用代码创建：基类为viewControllercell，有多少种cell，就创建多少个子类；基类创建方法loadwithdata：又子类去重写，传入参数为id类型的数据，子类可以改其类型为对应使用到的数据模型model；基类声明delegate，子类需要用到代理的，将代理方法声明都写到基类去

(4).创建自定义view，这个类用来封装页面中使用到的，不是cell的视图，如：cell中的一块view(如cell中有九宫格布局，每个格子是一个view内有多个子视图，为了便于

cell布局，将其封装起来，然后for循环创建九宫格)，基类为viewcontrollerview，基类创建方法loadwithdata：由子类去重写，传入参数为id类型的data，子类可以改其类型为对应使用到的数据模型model

(5).注意：一定要遵守一个页面一个列表的原则，避免列表嵌套列表，一个是内存消耗大，一个是controller逻辑复杂，不便代码维护

(6).注意：controller是领导，不做基层工作，什么意思呢？换句话说controller不能有时间判断、网络请求等具体逻辑。本人认为，正确的做法应该是：界面需要显示什么数据，controller让其下某个员工viewModel去取 -> viewModel发起网络请求得到model回调给controller -> controller拿到数据，ok，丢给view，告诉view：数据更新了，这是新数据，你拿去更新下界面把 -> view对新model进行拆解，得到目的数据进行界面更新(如果view嫌controller总是把整个model直接丢过来自己不能用还得做些繁琐的处理，那ok，再招一个打工仔viewModel，让他帮忙拆解)

本人页面编写习惯总结如下：

(1).controller的生命周期->(2).页面UI搭建及数据初始化(config函数)->(3).列表数据请求及数据刷新(refresh函数)->(4).各种Delegate实现，主要目的为了处理对应view层事件触发->(5).其他事件处理

注意：(1)、(2)相当于页面加载过程 (3)、(4)、(5)涉及数据请求业务逻辑处理等，应该将其主要实现交给viewModel层，整个(controller)主要负责(view)层事件监听->监听到事件触发后命令(viewModel)进行具体事件

处理，得到处理结果(Model)->将结果反馈给view层

C @implementation ViewController

- [-页面生命周期
- [M -dealloc
- [M -viewDidLoad
- [M -viewWillAppear:
- [M -viewDidAppear:
- [M -viewDidDisappear:
- [-页面UI搭建及数据初始化(Config函数)
- [M -configNavigationBar
- [M -configTable
- [M -configTimerManager
- [-列表数据请求及数据刷新(refresh函数)
- [M -refreshMessageNoReadNum
- [M -refreshTable
- [M -refreshSection:
- [-各种Delegate，主要目的是为了处理对应view层事件触发
- [-UITableViewDelegate
- [M -tableView:heightForRowAtIndexPath:
- [M -tableView:heightForFooterInSection:
- [M -tableView:heightForHeaderInSection:
- [M -tableView:viewForHeaderInSection:
- [-NavigationViewDelegate
- [M -navigationViewLeftDlegate
- [M -navigationViewReghtDlegate
- [-ViewControllerCellDelegate
- [M -viewControllerCell_Section0BtnClick:
- [M -viewControllerCell_Section1BtnClick:
- [M -viewControllerCell_Section2BtnClick:
- [M -bannerScrollView:didClickScrollView:
- [M -viewControllerCell_Section4BtnClick:
- [M -viewControllerCell_Section6BtnClick:
- [-ViewControllerViewDelegate
- [M -onClickSectionHeader:
- [-NSTimerManagerDelegate
- [M -timerDidFired:
- [-其他事件处理
- [M -tiaoZhuanZengZhiYeMian:withRoomId:withName:withroomId1:
- [M -tiaoZhuanCaiJingNeixianYeMian:withChannel_id:
- [M -tiaoZhuanKeHuZhuanXian
- [M -alertMessage:
- [M -handleEventAfterRequestIfYouKeNeedZengZhiDown
- [M -receiveNotificationForTuiChu:
- [M -receiveNotificationForChangeColor:
- [M -observeValueForKeyPath:ofObject:change:context:
- [M -didReceiveMemoryWarning