

## IOS 学习之第二个 View 使用 UITabBarController

前面有一篇博文 IOS 学习之 Tab Bar 的使用和视图切换 这是在 AppDelegate 里使用 Tabbar，这样的程序打开就是 TabbarView 了，有时候我们需要给程序做一些帮助页面，或者登录页面，之后才跳转到 tabbar View 里，或者后面的页面才使用 Tabbar 的，那这样怎么实现呢？

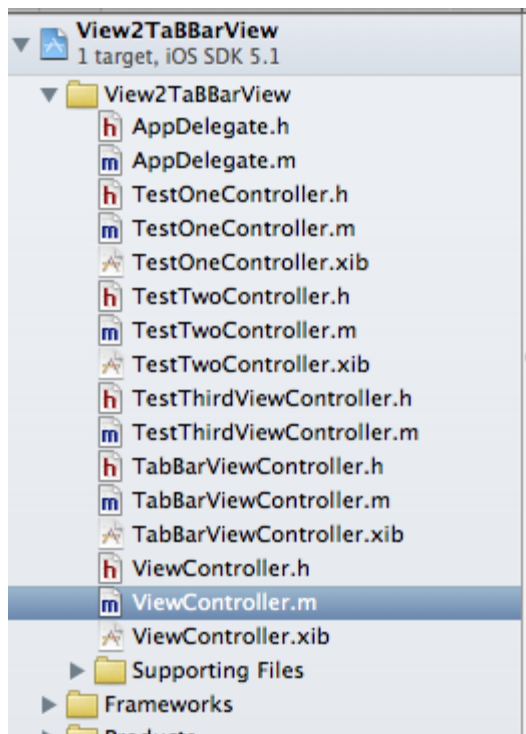
我们建立一个视图，然后在这个视图通过 `[self presentViewController : tabBar animated:YES]`；跳转来实现。

当程序中需要在多个 View 直接切换的时候，可以使用 UINavigationController，也可以用 ModalViewController。

UINavigationController 是通过导航条来切换多个 view。而如果 view 的数量比较少，且显示领域为全屏的时候，用 ModalViewController 就比较合适（比如需要用户输入信息的 view，结束后自动回复到之前的 view）

1、新建一个 Single View app，按 Command + N 新建三个 ViewController，都选上.xib 文件。

1.1 新建的 Controller 分别是：TestOneController TestTwoController TestThirdViewController，他们都继承 UIViewController。



单击 xib 文件，在.xib 文件的属性窗口里修改 View 的颜色，这样好在切换页面的时候区分出来是切换了页面。

好吧，我的 ThirdViewController 没有 xib, 可能是漏了，不过也没关系，一样能用。

## 1.2 添加 TabBarController

最重要的是再添加一个 TabBarController，这个需要继承的

UITabBarController

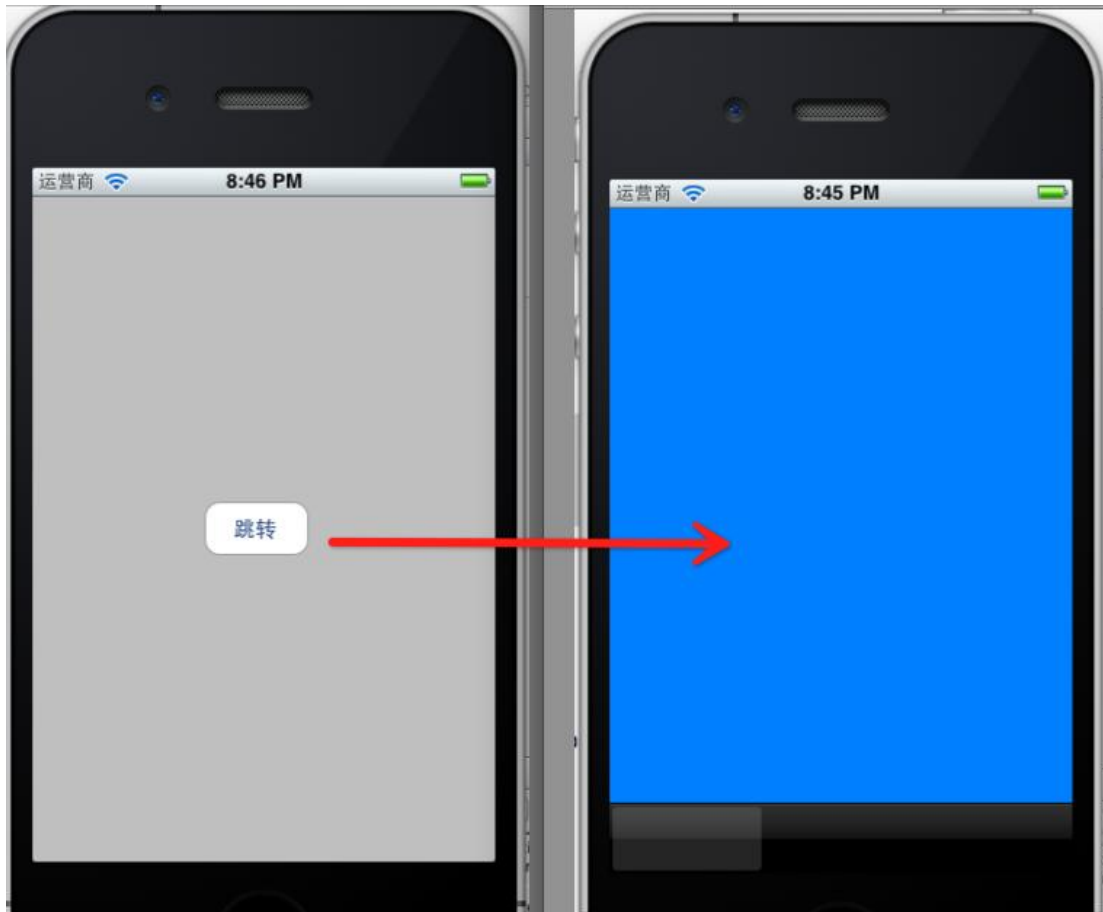
2、在 ViewController 也就是程序进来的第一个页面。在这里添加一个跳转的 Button, 并加上 Action, 然后在 Action 里实现跳转到 tabbar.

ViewController.m 里实现代码。这就跳转，把刚才建立的三个 ViewController 都添加到 Tabbar 里

[cpp] view plaincopy

```
1. - (IBAction)gotoTabbarView:(id)sender {
2.     NSMutableArray *items = [[NSMutableArray alloc] init];
3.     TestOneController *testOne1 = [[TestOneController alloc] in
        it];
4.     [items addObject:testOne1];
5.     TestTwoController *twoController = [[TestTwoController allo
        c] init];
6.     [items addObject:twoController];
7.     TestThirdViewController *thirdController = [[TestThirdViewC
        ontroller alloc] init];
8.     [items addObject:thirdController];
9.     // items 是数组，每个成员都是 UIViewController
10.    TabBarController *tabBar = [[TabBarController alloc
        ] init];
11.    [tabBar setTitle:@"TabBarController"];
12.    [tabBar setViewControllers:items];
13.
14.    [self presentViewController : tabBar animated:YES];
15.}
```

这样运行跳转到 TabbarView 里的了，但是现在的 tabbarView 里面的三个 Tab 按钮都空白的黑色。怎么添加图标和其他样式呢？



### 3、添加 UITabBarItem

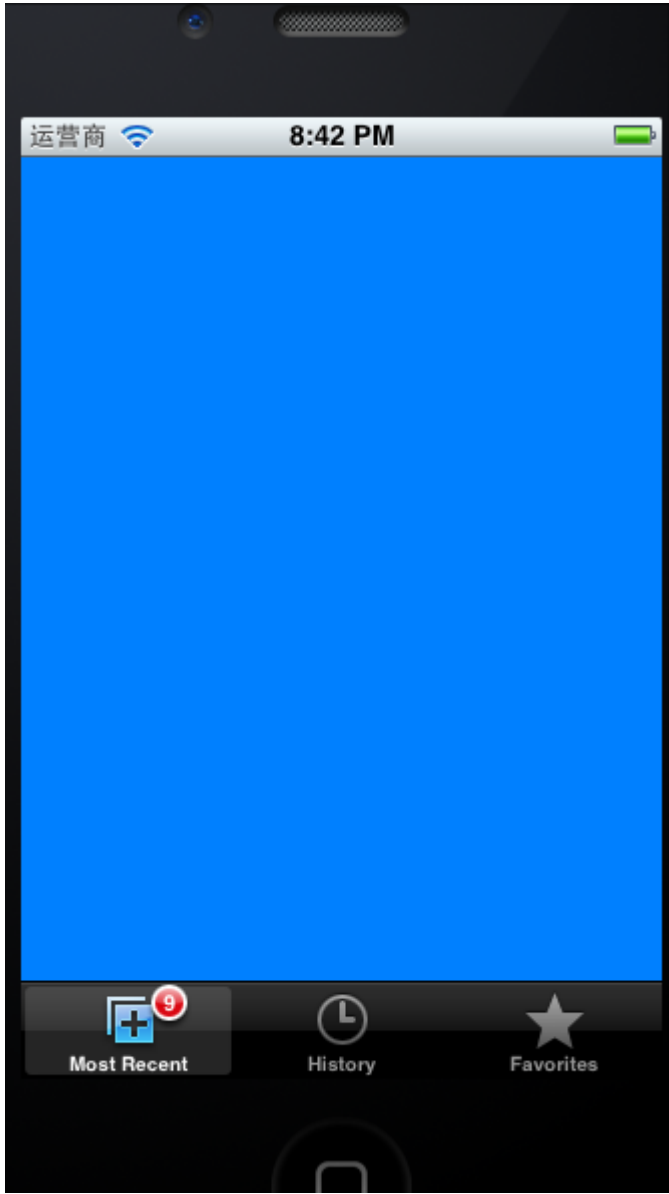
在三个 ViewController.m 文件里添加对应的 UITabBarItem，代码如下

[cpp] view plaincopy

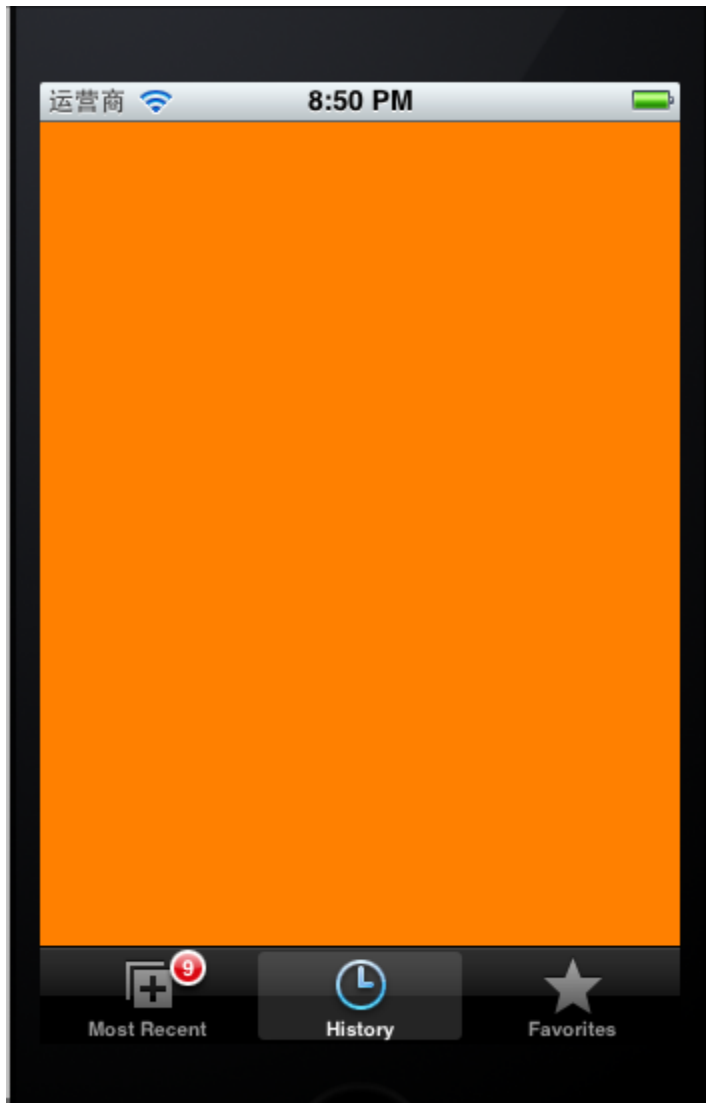
```
1. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
2. {
3.     self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
4.     if (self) {
5.         UITabBarItem *item = [[UITabBarItem alloc] initWithTabBarItemSystemItem:UITabBarItemSystemItemMostRecent tag:1];
6.         self.tabBarItem = item;
7.         self.tabBarItem.badgeValue = [NSString stringWithFormat:@"%d", 9];
8.     }
9.     return self;
10. }
```

UITabBarItemSystemItemMostRecent 这个 item 的风格常量，可以根据喜好改变。除此之外还可以用自定义的图片做为 item。这里就不演示自己添加图片的方式了。

`self.tabBarItem.badgeValue` 这个在 tabbar item 上显示泡泡的数字。  
对应的其他 ViewController 都添加上，tag 写不同的数字，后面要用到的。现在运行就有效果了



切换



#### 4、监听 Item 的点击事件

Tabbar 有了，怎么监听你点了哪个 item 呢？

实现 UITabBarDelegate。在 apple 的文档里查了一下，实现

- (void)tabBar:(UITabBar \*)tabBar didSelectItem:(UITabBarItem \*)item

这个方法即可监听

[cpp] view plaincopy

```
1. - (void)tabBar:(UITabBar *)tabBar didSelectItem:(UITabBarItem *)item
2. {
3.     if(item.tag == 1){
4.         NSLog(@"TestOneController");
5.     }else if(item.tag == 2){
6.         NSLog(@"TestTwoController");
7.     }else {
8.         NSLog(@"TestThirdController");
```

```
9.     }  
10. }
```

通过 tag 判断，运行切换：打印 log。

```
[cpp] view plaincopy
```

```
1. 2012-06-28 20:56:19.144 View2TaBBarView[5614:f803] TestTwoContr  
   oller  
2. 2012-06-28 20:56:19.785 View2TaBBarView[5614:f803] TestThirdCon  
   troller  
3. 2012-06-28 20:56:20.363 View2TaBBarView[5614:f803] TestTwoContr  
   oller  
4. 2012-06-28 20:56:20.843 View2TaBBarView[5614:f803] TestOneContr  
   oller
```

## 5、返回上个页面

通

过 [self.parentViewController dismissModalViewControllerAnimated:YES]; 返回上个页面

在 ThirdView 添加一个 button。添加 Action 事件。代码如下：

```
[cpp] view plaincopy
```

```
1. -(void)backAction:(id)sender{  
2.     [self.parentViewController dismissModalViewControllerAnimated:YES];  
3.  
4. }  
5. -(void)viewDidLoad  
6. {  
7.     [super viewDidLoad];  
8.     [self.view setBackgroundColor:[UIColor brownColor]];  
9.     UIButton *button = [[UIButton alloc] initWithFrame:CGRectMake(40, 50, 60, 30)];  
10.    [button setTitle:@"返回" forState:UIControlStateNormal];  
11.    [button addTarget:self action:@selector(backAction:) forControlEvents:UIControlEventTouchUpInside];  
12.    [self.view addSubview:button];  
13.    // Do any additional setup after loading the view.  
14. }
```

运行，点返回 button，返回了第一个页面了：

