MDMenuViewController

MDMenuViewController is an iOS container view controller that coordinates between its content view controllers through a side menu that is shown from the side with appealing animation , supports iOS 6.0 , 7.0 ,iPhone ,And iPad , Customisable in almost every aspect ,provided with documentation ,And a demo app to get you going right away.

should be compiled with ARC.

---

installation

- just drag the "MDMenuViewController classes" folder to your project

---

Basic How to

- prepare an NSArray of the content view Controllers

```
NSArray *viewControllers = [NSArray arrayWithObjects:[[ServicesViewController alloc]
initWithNibName:@"ServicesViewController" bundle:nil],[[FavouritViewController alloc]
initWithNibName:@"FavouritViewController" bundle:nil],[[ContactUsViewController alloc]
initWithNibName:@"ContactUsViewController" bundle:nil],[[AboutViewController alloc]
initWithNibName:@"AboutViewController" bundle:nil], nil];

MDMenuViewController *mainViewC = [[MDMenuViewController alloc]
initWithChildViewControllers:viewControllers];
```

- each content view controller maintained by MDMenuViewController must conform to the protocol MDMenuViewControllerDelegate And implementing two required methods

```
-(NSString*)titleForChildControllerMDMenuViewController:(MDMenuViewController*)menuController
```

which returns the title of the content view controller to be shown on the top bar and in the side menu

```
-(NSString*)iconForChildControllerMDMenuViewController:(MDMenuViewController*)menuController
```

which returns the image name in the bundle you want to show for the content view controller beside its title in side menu , if you wish not to add an icon just return an empty string.

and also declare a property

```
@property(nonatomic,weak)MDMenuViewController *MenuController
```

alternatively make the content view controllers inherit from MDMenuChildViewController ,and override the MDMenuViewControllerDelegate methods

- instantiate MDMenuViewController with the initialisation method

```
- (id)initWithChildViewControllers:(NSArray*)viewControllers
```

and pass it the NSArray of the content view Controllers

- show the MDMenuViewController instance.

- each content view controller can push other view controllers and pop them again like a UINaviagationController

each content view controller is a top view controller in a stack of its own so you can push on its stack other view

controllers.

```
[self.MenuController pushViewController:anotherViewController animated:YES];
```

you can also control the push transition animation , continue to see how .

- you can add a custom top bar left button by implemeting the MDMenuViewControllerDelegate Method

```
-(UIButton*)leftBarButtonForChildControllerMDMenuViewController:
(MDMenuViewController*)menuController;
```

and return a cutom UIButton to replace the left top bar button when this content view controller is shown

---

Customisation

Customise the menu Button

- you can change the icon of the menu button like the following

```
[mdMenuControllerInstance.topBar.menuBtn setImage:[UIImage imageNamed:@"side_menu.png"]
forState:UIControlStateNormal];

[mdMenuControllerInstance.topBar.menuBtn setImage:[UIImage imageNamed:@"side_menu_active.png"]
forState:UIControlStateHighlighted];
```

Customise the Back Button

- you can change the icon of the back button like the following

```
[mdMenuControllerInstance.topBar.backBtn setImage:[UIImage imageNamed:@"side_menu.png"]
forState:UIControlStateNormal];

[mdMenuControllerInstance.topBar.backBtn setImage:[UIImage imageNamed:@"side_menu_active.png"]
forState:UIControlStateHighlighted];
```

Customise the top bar

```
//****************************** MDMenuViewController customise top bar

// set main menu button image
[mdMenuViewControllerInstance.topBar.menuBtn setImage:[UIImage imageNamed:@"menu_icon.png"]
forState:UIControlStateNormal];



// set back button image
[mdMenuViewControllerInstance.topBar.backBtn setImage:[UIImage imageNamed:@"backBtn.png"]
forState:UIControlStateNormal];



// set top bar background color
[mdMenuViewControllerInstance.topBar setBackgroundColor:[UIColor colorWithWhite:0.85 alpha:1.0]];



// set top bar title text color
[mdMenuViewControllerInstance.topBar.titleLbl setTextColor:[UIColor whiteColor]];



// set top bar background image
mdMenuViewControllerInstance.topBar.backgroundImage.image = [UIImage imageNamed:@"topBar.jpg"];



//*********************************************************************************************
```

Customise the side Menu

```
//****************************** MDMenuViewController customise menu view
```

```
// menu item title text color when selected
[mdMenuViewControllerInstance.menuView setMenuItemTitleTextColor:[UIColor whiteColor]
forState:UIControlStateHighlighted];


// menu item title text color when unSelected
[mdMenuViewControllerInstance.menuView setMenuItemTitleTextColor:[UIColor whiteColor]
forState:UIControlStateNormal];


// menu item background color when unSelected
[mdMenuViewControllerInstance.menuView setMenuItemBackgroundColor:[UIColor colorWithRed:(46.0f/
255.0f) green:(46.0f/255.0f) blue:(46.0f/255.0f) alpha:1.0] forState:UIControlStateNormal];


// menu item background color when selected
[mdMenuViewControllerInstance.menuView setMenuItemBackgroundColor:[UIColor colorWithRed:(47.0f/
255.0f) green:(123.0f/255.0f) blue:(154.0f/255.0f) alpha:1.0] forState:UIControlStateHighlighted];


// menu vie background color
[mdMenuViewControllerInstance.menuView setBackgroundColor:[UIColor colorWithRed:(46.0f/255.0f)
green:(46.0f/255.0f) blue:(46.0f/255.0f) alpha:1.0]];



//disabel ripple animation

[mdMenuViewControllerInstance.menuView setDisableRippleAnimation:NO];


//**************************************************************************************************
```

- if you want full control over the content of the side menu view and how its shown , you can create your own menu view provided that it inherits from the base MenuView class and override its interface methods - the base MenuView class is not meant to be used on its own , its just meant to be sub classed - then set your own custom MenuView to the menuView property of your MDMenuViewController instance.

```
mdMenuControllerInstance.menuView = yourCustomMenuView;
```

Customise transition Animations

- MDMenuViewController has default animations for push and pop actions , you can make your own animations by declaring your own class that conform to the protocol MDTransitionAnimatorProtocol with only one required method

```
-(void)transitionInView:(UIView*)containerView fromView:(UIView*)fromView toView:(UIView*)toView
```

Then to use your custom animator call this method when pushing viewControllers

```
[self.MenuController pushViewController:anotherController
withTransitionAnimator:yourCustomAnimator];
```

In order to control pop animation just implement the MDMenuViewControllerDelegate method in your content View Controller

```
-(void)topbarViewDidTapBackButton;
```

And in the implementation call

```
[self.MenuController popViewControllerWithTransitionAnimator:youCustomAnimator];
```

In general you can show any viewController using the following method

```
[self.MenuController showViewController:viewController withTransitionAnimator:yourCustomAnimator];
```