

iOS 界面之间传值总结



作者 王小宾 (/u/098b303fcf0e) + 关注 (/sign_in)

2015.11.10 17:41 字数 1243 阅读 2717 评论 10 喜欢 6

(/u/098b303fcf0e)

iOS页面间传值的方式

(NSUserDefaults/Delegate/NSNotification/Block/单例)

实现了以下iOS页面间传值：1.委托delegate方式；2.通知notification方式；3.block方式；4.UserDefault或者文件方式；5.单例模式方式；6.通过设置属性，实现页面间传值
在iOS开发中，我们经常会遇到页面间跳转传值的问题，现归纳总结一下：

情况1：A页面跳转到B页面

方法：

在B页面的控制器中，编写对应的属性，在A页面跳转到B页面的地方，给B的属性赋值即可

1.

//SecondViewController.h

```
@property (nonatomic) NSInteger flag;//当前系统标示 (0: 其他传值方式; 1: block传值方式)
```

在A页面的试图控制器中

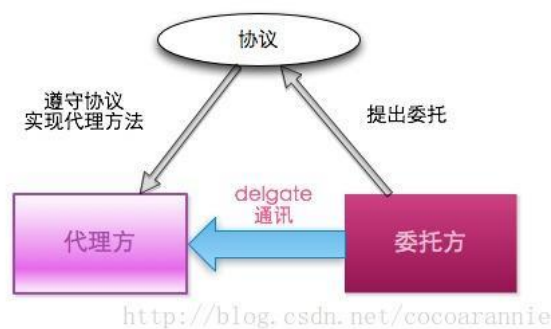
1.//RootViewController.m

```
- (IBAction)showSecondView:(id)sender {
    SecondViewController *second = [[SecondViewController alloc] initWithNibName:@"Se
    second.delegate = self;
    second.flag = 0;
    [self presentViewController:second animated:YES completion:nil];
}
```

情况2：A页面跳转到B页面，B页面再跳转回A页面

主流方案：

(1) 通过委托delegate的方式实现



//SecondViewController.h

```
@protocol secondViewDelegate
-(void)showName:(NSString *)nameString;
@end
```

设置代理(为防止循环引用, 此处采用了weak)

//SecondViewController.h

```
@interface SecondViewController : UIViewController
@property (nonatomic, weak)id<secondViewDelegate> delegate;
@property (nonatomic, copy) ablock block;
@end
```

//SecondViewController.m

```
- (IBAction)delegateMethod:(id)sender {
if([self notEmpty]) {
[self.delegate showName:self.nameTextField.text];
[self dismissViewControllerAnimated:YES completion:nil];
}
else
{
[self showAlert];
}
}
```

//RootViewController.m

```
-(void)showName:(NSString *)nameString{
self.nameLabel.text = nameString;
}
```

最重要也是最容易忽略的, 就是一定要设置delegate的指向。(2) 通过notification的方式实现

在B页面的控制器中, 发送通知:

//SecondViewController.m

```
- (IBAction)notificationMethod:(id)sender {
if ([self notEmpty]) {
[[NSNotificationCenter defaultCenter] postNotificationName:
@'ChangeNameNotification' object:self userInfo:@{@"name":self.nameTextField.text}];
[self dismissViewControllerAnimated:YES completion:nil];
}else{
[self showAlert];
}
}
```

在A页面的控制器中, 注册通知:

//RootViewController.m

```
- (void)viewDidLoad{
[super viewDidLoad];
// Do any additional setup after loading the view from its nib.
[[NSNotificationCenter defaultCenter] addObserver:self selector:
@selector
(changeNameNotification:) name:@
'ChangeNameNotification'
object:nil];
}
```

当我们不使用时，要记得删掉通知：

//RootViewController.m

```
-(void)dealloc{
    [[NSNotificationCenter defaultCenter] removeObserver:self];
}
```

调用，显示

//RootViewController.m

```
-(void)ChangeNameNotification:(NSNotification*)notification{
    NSDictionary *nameDictionary = [notification userInfo];
    self.nameLabel.text = [nameDictionary objectForKey:@"name"];
}
```

(3) block方式实现

block介绍:<http://blog.csdn.net/totogo2010/article/details/7839061>

(<http://blog.csdn.net/totogo2010/article/details/7839061>)

链接一篇描述block回调挺有意思的文章：

<http://blog.csdn.net/mobanchengshuang/article/details/11751671>

(<http://blog.csdn.net/mobanchengshuang/article/details/11751671>)

分析：

在B视图控制器中，定义一个block，参数为字符串

//SecondViewController.h

```
typedef void (^ablock)(NSString *str);
```

//SecondViewController.h

```
@property (nonatomic, copy) ablock block;
```

在B视图控制器中，当输入名字，点击对应的确定按钮后

```
- (IBAction)blockMethod:(id)sender {
    if ([self notEmpty]) {
        if(self.block) {
            self.block(self.nameTextField.text);
            [self dismissViewControllerAnimated:YES completion:nil];
        }
    }else{
        [self showAlert];
    }
}
```

在A视图显示，回调block

```
- (IBAction)showSecondWithBlock:(id)sender {
    SecondViewController *second = [[SecondViewController alloc] initWithNibName:@"SecondViewController" bundle:nil];
    [self presentViewController:second animated:YES completion:nil];
    second.block = ^(NSString *str){
        self.nameLabel.text = str;
    };
}
```

在查阅资料的过程中，我还看到了以下几种方案：

- (1) 使用SharedApplication,定义一个变量来传递（感觉和单例的方式一样）
- (2) 使用文件，或者NSUserDefaults来传递

//通过文件或者UserDefaults方式存值(感觉不太适合此类传值，如果要用文件或者UserDefaults方式存值的话，可以考虑此方式)

```
- (IBAction)userDefaultMethod:(id)sender {
    if ([self notEmpty]) {
        [[NSUserDefaults standardUserDefaults] setObject:self.nameTextField.text forKey:@"myNameText"];
    }
    [self dismissViewControllerAnimated:YES completion:nil];
} else {
    [self showAlert];
}
```

在A视图控制器显示

```
-(void)viewDidAppear:(BOOL)animated{
    [super viewDidAppear:animated];
    //如果想测试通过UserDefaults方式传值或者通过单例方式传值，取消以下注释即可
    /*
    if ([[NSUserDefaults standardUserDefaults] objectForKey:@"myNameText"] length] !=
    self.nameLabel.text = [[NSUserDefaults standardUserDefaults] objectForKey:@"myNameText"];
    [[NSUserDefaults standardUserDefaults] setObject:@"myNameText" forKey:@"myNameText"];
    */
    DataSource *dataSource = [DataSource sharedDataSource];
    if ([dataSource.myName length] != 0) {
        self.nameLabel.text = dataSource.myName;
        dataSource.myName = @"myName";
    }
    /*
    */
}
```

- (3) 通过一个单例的class来传递

B视图控制器

//通过单例方式传值(感觉不太适合此类传值，如果要用单例方式传值的话，可以考虑此方式)

```
- (IBAction)singletonMethod:(id)sender {
    if ([self notEmpty]) {
        DataSource *dataSource = [DataSource sharedDataSource];
        dataSource.myName = self.nameTextField.text;
        [self dismissViewControllerAnimated:YES completion:nil];
    } else {
        [self showAlert];
    }
}
```

A视图控制器显示

```

-(void)viewDidAppear:(BOOL)animated{
    [super viewDidAppear:animated];
    //如果想测试通过UserDefaults方式传值或者通过单例方式传值，取消以下注释即可
    /*
    if ([[NSUserDefaults standardUserDefaults] objectForKey:@"myNameText"] length] != 0) {
        self.nameLabel.text = [[NSUserDefaults standardUserDefaults] objectForKey:@"myNameText"];
        [[NSUserDefaults standardUserDefaults setObject:@" " forKey:@"myNameText"];
    }
    DataSource *dataSource = [DataSource sharedInstance];
    if ([dataSource.myName length] != 0) {
        self.nameLabel.text = dataSource.myName;
        dataSource.myName = @" ";
    }
    */
}
@end

```

这里面用到了单例模式，编写了DataSource这个类，存放数据

```

#import <Foundation/Foundation.h>
@interface DataSource : NSObject
@property (nonatomic, strong) NSString *myName;
+(DataSource*)sharedDataSource;
@end

```

```

@implementation DataSource
+(DataSource *)sharedDataSource{
    static DataSource *dataSource = nil;
    static dispatch_once_t once;
    dispatch_once(&once, ^{
        dataSource = [DataSource new];
    });
    return dataSource;
}
@end

```

程序运行截图

A视图:



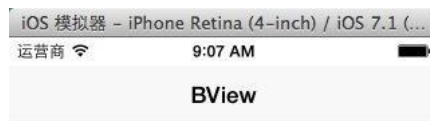
showSecond

showSecondWithBlock

Name

Button

B视图



input your name

Cancel

delegateMethod

notificationMethod

userDefaultMethod

singletonMethod

当输入姓名，并点击对应的确认按钮后，会回到A视图，并显示在B视图中输入的姓名



showSecond

showSecondWithBlock

JuneWang



祝：玩得开心，有什么别的办法或者不正确的地方，欢迎指正。

📖 日记本 (/nb/2270992) [举报文章](#) © 著作权归作者所有

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

打赏支持

🤍 喜欢 (/sign_in) | 6



(<http://cwb.assets.jianshu.io/notes/images/2379217/>)



登录 (/sign_in) 发表评论

10条评论 [只看作者](#)

[按时间正序](#) [按时间倒序](#) [按喜欢排序](#)



Joy__ (/u/9c51a213b02e)
2楼 · 2016.01.05 13:50
(/u/9c51a213b02e)
那个比较有优势呢？



武林盟主和穿山甲 (/u/73885ac131e2)

3楼 · 2016.09.27 18:32

(/u/73885ac131e2)

两个没有关系的页面怎么传值呢？



王小宾 (/u/098b303fcf0e): @武林盟主和穿山甲 (/users/73885ac131e2) 两个没有任何关系的界面为什么要发生传值呢？

2016.09.27 18:34 回复 (/sign_in)

武林盟主和穿山甲 (/u/73885ac131e2): @王小宾 (/users/098b303fcf0e) 比如说缓存一个视屏。。A页面观看页面点击下载，在B页面下载并显示下载进度。我现在是用通知来处理，但是有问题，如果不先进入B页面，A页面发送的通知B页面不会进入那个方法。

2016.09.28 12:16 回复 (/sign_in)

武林盟主和穿山甲 (/u/73885ac131e2): @王小宾 (/users/098b303fcf0e) 诶。。我做出来了，总之谢谢你啦

2016.09.28 14:19 回复 (/sign_in)

添加新评论 (/sign_in) | 还有3条评论，[展开查看](#)



空调好刺激啊 (/u/52d630e79946)

4楼 · 2016.10.27 16:05

(/u/52d630e79946)

楼主，，，问下。。通过push跳转超过2个vc间如何传值呢？比如说A->B->C C直接返回到A如何将C的值传回到A呢？



王小宾 (/u/098b303fcf0e): @空调好刺激啊 (/users/52d630e79946) 可以通过通知或者block，或者代理

2016.10.31 13:58 回复 (/sign_in)

添加新评论 (/sign_in)

被以下专题收入，发现更多相似内容

