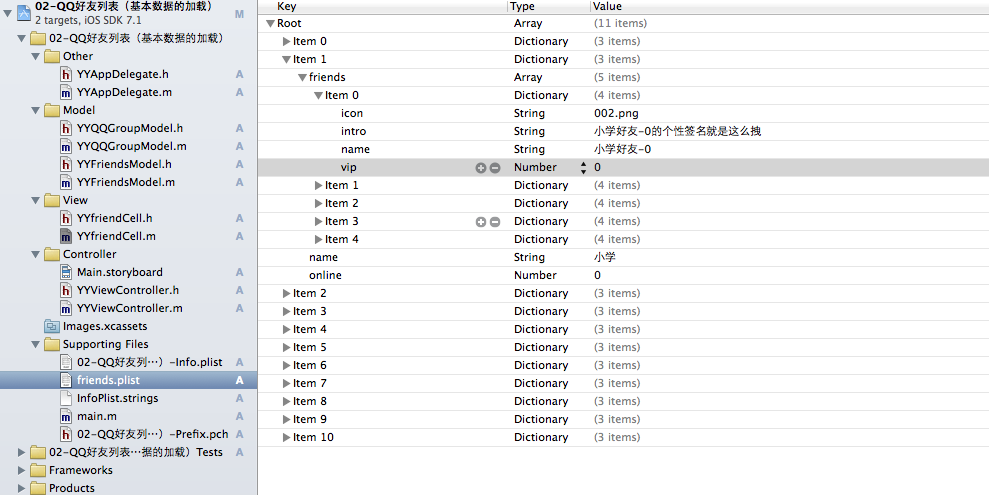
**iOS开发UI篇—使用UItableview完成一个简单的QQ好友列表（一）**

**一、项目结构和plist文件**

****

二、实现代码

1.说明：

主控制器直接继承UITableViewController

[](javascript:void(0);)

// YYViewController.h

// 02-QQ好友列表（基本数据的加载）

//

// Created by apple on 14-5-31.

// Copyright (c) 2014年 itcase. All rights reserved.

//

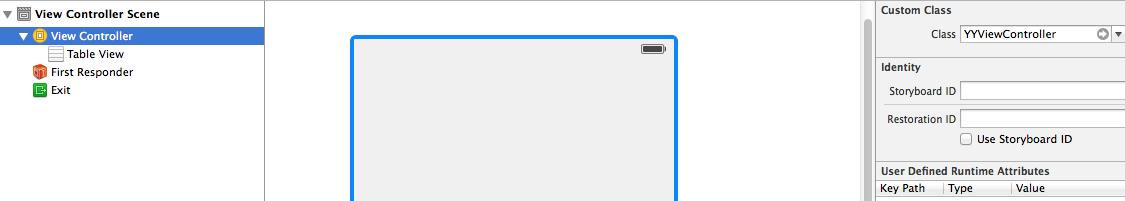
#import <UIKit/UIKit.h>

@interface YYViewController : UITableViewController

@end

[](javascript:void(0);)

在storyboard中进行了关联



2.代码

数据模型部分：

YYQQGroupModel.h文件

[](javascript:void(0);)

1 //

2 // YYQQGroupModel.h

3 // 02-QQ好友列表（基本数据的加载）

4 //

5 // Created by apple on 14-5-31.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import <Foundation/Foundation.h>

10

11 @interface YYQQGroupModel : NSObject

12 /\*\*

13 \* 名称属性

14 \*/

15 @property(nonatomic,copy)NSString \*name;

16 /\*\*

17 \* 是否在线

18 \*/

19 @property(nonatomic,copy)NSString \*online;

20 /\*\*

21 \* 好友列表

22 \*/

23 @property(nonatomic,strong)NSArray \*friends;

24

25 -(instancetype)initWithDict:(NSDictionary \*)dict;

26 +(instancetype) qqGroupModelWithDict:(NSDictionary \*)dict;

27 @end

[](javascript:void(0);)

YYQQGroupModel.m文件

[](javascript:void(0);)

1 //

2 // YYQQGroupModel.m

3 // 02-QQ好友列表（基本数据的加载）

4 //

5 // Created by apple on 14-5-31.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import "YYQQGroupModel.h"

10 #import "YYFriendsModel.h"

11

12 @implementation YYQQGroupModel

13 -(instancetype)initWithDict:(NSDictionary \*)dict

14 {

15 if (self=[super init]) {

16 //将字典转换为模型

17 [self setValuesForKeysWithDictionary:dict];

18

19 //定义一个数组来保存转换后的模型

20 NSMutableArray \*models=[NSMutableArray arrayWithCapacity:self.friends.count];

21 for (NSDictionary \*dict in self.friends) {

22 YYFriendsModel \*friends=[YYFriendsModel friendsWithDict:dict];

23 [models addObject:friends];

24 }

25 \_friends=[models copy];

26 }

27 return self;

28 }

29

30 +(instancetype)qqGroupModelWithDict:(NSDictionary \*)dict

31 {

32 return [[self alloc]initWithDict:dict];

33 }

34 @end

[](javascript:void(0);)

YYFriendsModel.h文件

[](javascript:void(0);)

1 //

2 // YYFriendsModel.h

3 // 02-QQ好友列表（基本数据的加载）

4 //

5 // Created by apple on 14-5-31.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import <Foundation/Foundation.h>

10

11 @interface YYFriendsModel : NSObject

12 /\*\*

13 \* 每个好友的名称

14 \*/

15 @property(nonatomic,copy)NSString \*name;

16 /\*\*

17 \*每个好友的头像

18 \*/

19 @property(nonatomic,copy)NSString \*icon;

20 /\*\*

21 \* 每个好友的个性签名

22 \*/

23 @property(nonatomic,copy)NSString \*intro;

24 /\*\*

25 \* 该好友是否是vip

26 \*/

27 @property(nonatomic,assign,getter = isVip)BOOL vip;

28

29 -(instancetype)initWithDict:(NSDictionary \*)dict;

30 +(instancetype)friendsWithDict:(NSDictionary \*)dict;

31 @end

[](javascript:void(0);)

YYFriendsModel.m文件

[](javascript:void(0);)

1 //

2 // YYFriendsModel.m

3 // 02-QQ好友列表（基本数据的加载）

4 //

5 // Created by apple on 14-5-31.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import "YYFriendsModel.h"

10

11 @implementation YYFriendsModel

12 -(instancetype)initWithDict:(NSDictionary \*)dict

13 {

14 if (self=[super init]) {

15 [self setValuesForKeysWithDictionary:dict];

16 }

17 return self;

18 }

19

20 +(instancetype)friendsWithDict:(NSDictionary \*)dict

21 {

22 return [[self alloc]initWithDict:dict];

23 }

24 @end

[](javascript:void(0);)

视图部分

YYfriendCell.h文件

[](javascript:void(0);)

1 //

2 // YYfriendCell.h

3 // 02-QQ好友列表（基本数据的加载）

4 //

5 // Created by apple on 14-5-31.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import <UIKit/UIKit.h>

10 @class YYFriendsModel;

11 @interface YYfriendCell : UITableViewCell

12

13 @property(nonatomic,strong)YYFriendsModel \*friends;

14

15 +(instancetype)cellWithTableview:(UITableView \*)tableView;

16 @end

[](javascript:void(0);)

YYfriendCell.m文件

[](javascript:void(0);)

1 //

2 // YYfriendCell.m

3 // 02-QQ好友列表（基本数据的加载）

4 //

5 // Created by apple on 14-5-31.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import "YYfriendCell.h"

10 #import "YYFriendsModel.h"

11 //私有扩展

12 @interface YYfriendCell()

13

14

15 @end

16 @implementation YYfriendCell

17

18 +(YYfriendCell \*)cellWithTableview:(UITableView \*)tableView

19 {

20 static NSString \*identifier=@"qq";

21 YYfriendCell \*cell=[tableView dequeueReusableCellWithIdentifier:identifier];

22 if (cell==nil) {

23 //这里使用系统自带的样式

24 cell=[[YYfriendCell alloc]initWithStyle:UITableViewCellStyleSubtitle reuseIdentifier:identifier];

25 NSLog(@"创建一个cell");

26 }

27 return cell;

28 }

29

30 -(void)setFriends:(YYFriendsModel \*)friends

31 {

32 \_friends=friends;

33 //1.设置头像

34 self.imageView.image=[UIImage imageNamed:\_friends.icon];

35 //2.设置昵称

36 self.textLabel.text=\_friends.name;

37 //3.设置简介

38 self.detailTextLabel.text=\_friends.intro;

39 //判断是否是会员

40 /\*\*

41 \* 这里有个注意点，如果不写else设置为黑色，会怎么样？

42 \*/

43 if (\_friends.isVip) {

44 [self.textLabel setTextColor:[UIColor redColor]];

45 }else

46 {

47 [self.textLabel setTextColor:[UIColor blackColor]];

48 }

49 }

50 @end

[](javascript:void(0);)

主控制器部分

YYViewController.m文件

[](javascript:void(0);)

1 //

2 // YYViewController.m

3 // 02-QQ好友列表（基本数据的加载）

4 //

5 // Created by apple on 14-5-31.

6 // Copyright (c) 2014年 itcase. All rights reserved.

7 //

8

9 #import "YYViewController.h"

10 #import "YYQQGroupModel.h"

11 #import "YYfriendCell.h"

12 #import "YYFriendsModel.h"

13

14 @interface YYViewController ()

15 /\*\*

16 \* 用来保存所有的分组数据

17 \*/

18 @property(nonatomic,strong)NSArray \*groupFriends;

19 @end

20

21 @implementation YYViewController

22 #pragma mark-懒加载

23 //1.先拿到数据，实现懒加载

24 -(NSArray \*)groupFriends

25 {

26 if (\_groupFriends==nil) {

27 NSString \*fullpath=[[NSBundle mainBundle]pathForResource:@"friends.plist" ofType:nil];

28 NSArray \*arrayM=[NSArray arrayWithContentsOfFile:fullpath];

29

30 NSMutableArray \*models=[NSMutableArray arrayWithCapacity:arrayM.count];

31 for (NSDictionary \*dict in arrayM) {

32 YYQQGroupModel \*group=[YYQQGroupModel qqGroupModelWithDict:dict];

33 [models addObject:group];

34 }

35 \_groupFriends=[models copy];

36 }

37 return \_groupFriends;

38 }

39

40 - (void)viewDidLoad

41 {

42 [super viewDidLoad];

43 self.tableView.sectionHeaderHeight = 100;

44 }

45

46 #pragma mark- 设置数据源

47 //返回多少组

48 //为什么这里不会智能提示？因为这些方法是uitableview协议里的，默认并没有遵守协议，让主控制器类继承uitableviewcontroller后，就已经遵守了

49 -(NSInteger)numberOfSectionsInTableView:(UITableView \*)tableView

50 {

51 return self.groupFriends.count;

52 }

53 //每组返回多少行

54 -(NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section

55 {

56 //取出对应的组模型

57 YYQQGroupModel \*group=self.groupFriends[section];

58 //返回对应组中的好友数

59 return group.friends.count;

60 }

61 //每组每行的内容

62 -(UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

63 {

64 //1.创建cell

65 YYfriendCell \*cell=[YYfriendCell cellWithTableview:tableView];

66

67 //2.设置cell

68 YYQQGroupModel \*group=self.groupFriends[indexPath.section];

69 YYFriendsModel \*friends=group.friends[indexPath.row];

70 cell.friends=friends;

71 //3.返回一个cell

72 return cell;

73 }

74

75

76 #pragma mark - 代理方法

77 // 当一个分组标题进入视野的时候就会调用该方法

78 - (UIView \*)tableView:(UITableView \*)tableView viewForHeaderInSection:(NSInteger)section

79 {

80 // 1.创建头部视图

81 UIView \*view = [[UIView alloc] init];

82 view.backgroundColor = [UIColor grayColor];

83 // 2.返回头部视图

84 return view;

85 }

86

87 //设置分组头部标题的高度

88 -(CGFloat)tableView:(UITableView \*)tableView heightForHeaderInSection:(NSInteger)section

89 {

90 return 44;

91 }

92

93 #pragma mark 隐藏状态栏

94 -(BOOL)prefersStatusBarHidden

95 {

96 return YES;

97 }

98 @end

[](javascript:void(0);)

实现的简陋效果：



三、注意点

（1）设置头部视图的方法



（2）在重写set方法时，应该考虑到回滚。

