一、实现效果



点击加载更多按钮，出现一个加载图示，三秒钟后添加两条新的数据。

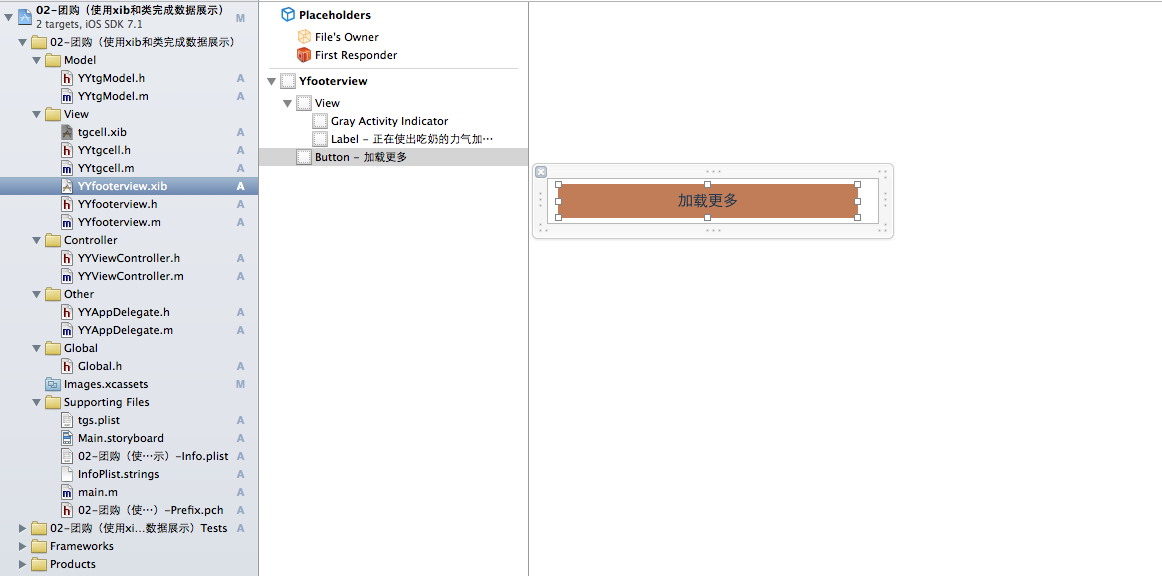
二、实现代码和说明

当在页面（视图部分）点击加载更多按钮的时候，主页面（主控制器）会加载两条数据进来。

视图部分的按钮被点击的时候，要让主控制器加载数据，刷新表格，2B青年会在视图中增加一个主控制器的属性，通过这个属性去调用进行加载，但在开发中通常通过代理模式来完成这个操作。

下面分别是两种实现的代码。

1、项目结构和说明



说明：加载更多永远都放在这个tableview的最下端，因此这里设置成了这个tableview的tableFooterView。

 self.tableview.tableFooterView=footerview;

在实现上通过xib来进行处理，考虑到左右的留白，以及点击后的要切换到加载按钮和文字，要同时控制图标和文字，因此把加载图标和文字提示放在了一个view中以便控制，这个xib已经和YYfooterview.xib进行了关联，通过这个类来控制xib。

2、实现代码

（1）.垃圾代码

数据模型部分

YYtg.h文件

[](javascript:void(0);)

1 //

2 // YYtg.h

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import <Foundation/Foundation.h>

10 #import "Global.h"

11

12 @interface YYtgModel : NSObject

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

14 @property(nonatomic,copy)NSString \*buyCount;

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

16 @property(nonatomic,copy)NSString \*price;

17

18 //对外接口

19 YYinitH(tg)

20 @end

[](javascript:void(0);)

YYtg.m文件

[](javascript:void(0);)

1 //

2 // YYtg.m

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import "YYtgModel.h"

10

11 @implementation YYtgModel

12 YYinitM(tg)

13 @end

[](javascript:void(0);)

注意：对于数据转模型部分的构造方法接口和实现代码已经通过自定义带参数的宏来进行了封装。

封装代码如下：

[](javascript:void(0);)

1 #ifndef \_0\_\_\_\_\_\_\_\_\_\_\_\_Global\_h

2 #define \_0\_\_\_\_\_\_\_\_\_\_\_\_Global\_h

3

4 /\*\*

5 \* 自定义带参数的宏

6 \*/

7 #define YYinitH(name) -(instancetype)initWithDict:(NSDictionary \*)dict;\

8 +(instancetype)name##WithDict:(NSDictionary \*)dict;

9

10

11 #define YYinitM(name) -(instancetype)initWithDict:(NSDictionary \*)dict\

12 {\

13 if (self=[super init]) {\

14 [self setValuesForKeysWithDictionary:dict];\

15 }\

16 return self;\

17 }\

18 \

19 +(instancetype)name##WithDict:(NSDictionary \*)dict\

20 {\

21 return [[self alloc]initWithDict:dict];\

22 }\

23

24 #endif

[](javascript:void(0);)

视图部分

YYtgcell.h文件

[](javascript:void(0);)

1 //

2 // YYtgcell.h

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import <UIKit/UIKit.h>

10 #import "YYtgModel.h"

11

12 @interface YYtgcell : UITableViewCell

13 @property(nonatomic,strong)YYtgModel \*yytg;

14

15 //把加载数据（使用xib创建cell的内部细节进行封装）

16 +(instancetype)tgcellWithTableView:(UITableView \*)tableView;

17 @end

[](javascript:void(0);)

YYtgcell.m文件

[](javascript:void(0);)

1 //

2 // YYtgcell.m

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import "YYtgcell.h"

10 //私有扩展

11 @interface YYtgcell()

12 @property (strong, nonatomic) IBOutlet UIImageView \*img;

13 @property (strong, nonatomic) IBOutlet UILabel \*titlelab;

14 @property (strong, nonatomic) IBOutlet UILabel \*pricelab;

15 @property (strong, nonatomic) IBOutlet UILabel \*buycountlab;

16 @end

17 @implementation YYtgcell

18

19 #pragma mark 重写set方法，完成数据的赋值操作

20 -(void)setYytg:(YYtgModel \*)yytg

21 {

22 \_yytg=yytg;

23 self.img.image=[UIImage imageNamed:yytg.icon];

24 self.titlelab.text=yytg.title;

25 self.pricelab.text=[NSString stringWithFormat:@"$%@",yytg.price];

26 self.buycountlab.text=[NSString stringWithFormat:@"已有%@人购买",yytg.buyCount];

27 }

28

29 +(instancetype)tgcellWithTableView:(UITableView \*)tableView

30 {

31 static NSString \*identifier= @"tg";

32 YYtgcell \*cell=[tableView dequeueReusableCellWithIdentifier:identifier];

33 if (cell==nil) {

34 //如何让创建的cell加个戳

35 //对于加载的xib文件，可以到xib视图的属性选择器中进行设置

36 cell=[[[NSBundle mainBundle]loadNibNamed:@"tgcell" owner:nil options:nil]firstObject];

37 NSLog(@"创建了一个cell");

38 }

39 return cell;

40 }

41

42 @end

[](javascript:void(0);)

YYfooterview.h文件

[](javascript:void(0);)

1 //

2 // YYfooterview.h

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import <UIKit/UIKit.h>

10 @class YYViewController;

11 @interface YYfooterview : UIView

12 @property(nonatomic,strong) YYViewController \*controller;

13 @end

[](javascript:void(0);)

YYfooterview.m文件

[](javascript:void(0);)

1 //

2 // YYfooterview.m

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import "YYfooterview.h"

10 #import "YYViewController.h"

11

12 @interface YYfooterview ()

13 @property (strong, nonatomic) IBOutlet UIActivityIndicatorView \*loadingview;

14 @property (strong, nonatomic) IBOutlet UIButton \*loadbtn;

15

16 @end

17 @implementation YYfooterview

18 - (IBAction)loadbtclick {

19 NSLog(@"按钮被点击了");

20 //隐藏按钮

21 self.loadbtn.hidden=YES;

22 //显示菊花

23 self.loadingview.hidden=NO;

24

25 #warning 模拟发送网络请求, 3秒之后隐藏菊花

26 dispatch\_after(dispatch\_time(DISPATCH\_TIME\_NOW, (int64\_t)(3.0 \* NSEC\_PER\_SEC)), dispatch\_get\_main\_queue(), ^{

27 // 3.调用控制的加载数据方法

28 [self.controller LoadMore];

29 // 4.隐藏菊花视图

30 self.loadingview.hidden = YES;

31 // 5.显示按钮

32 self.loadbtn.hidden = NO;

33 });

34 }

35

36 @end

[](javascript:void(0);)

主控制器

YYViewController.h文件

[](javascript:void(0);)

1 //

2 // YYViewController.h

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import <UIKit/UIKit.h>

10

11 @interface YYViewController : UIViewController

12 //公开接口

13 //- (void)LoadMore;

14 @end

[](javascript:void(0);)

YYViewController.m文件

[](javascript:void(0);)

1 //

2 // YYViewController.m

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import "YYViewController.h"

10 #import "YYtgModel.h"

11 #import "YYtgcell.h"

12 #import "YYfooterview.h"

13

14 @interface YYViewController ()<UITableViewDataSource,UITableViewDelegate>

15 @property (strong, nonatomic) IBOutlet UITableView \*tableview;

16

17 @property(strong,nonatomic)NSMutableArray \*tg;

18 @end

19

20 @implementation YYViewController

21

22 #pragma mark-加载数据方法

23 -(void)LoadMore

24 {

25 //创建模型

26 YYtgModel \*tgmodel=[[YYtgModel alloc]init];

27 tgmodel.title=@"菜好上桌";

28 tgmodel.icon=@"5ee372ff039073317a49af5442748071";

29 tgmodel.buyCount=@"20";

30 tgmodel.price=@"10000";

31 //将模型添加到数组中

32 [self.tg addObject:tgmodel];

33

34 YYtgModel \*tgmodelq=[[YYtgModel alloc]init];

35 tgmodelq.title=@"菜好上桌1";

36 tgmodelq.icon=@"5ee372ff039073317a49af5442748071";

37 tgmodelq.buyCount=@"20";

38 tgmodelq.price=@"10000";

39

40 [self.tg addObject:tgmodelq];

41 //刷新表格

42 [self.tableview reloadData];

43 }

44

45 - (void)viewDidLoad

46 {

47 [super viewDidLoad];

48 self.tableview.rowHeight=80.f;

49

50 //加载底部视图

51 //从xib中获取数据

52 UINib \*nib=[UINib nibWithNibName:@"YYfooterview" bundle:nil];

53 YYfooterview \*footerview=[[nib instantiateWithOwner:nil options:nil] firstObject];

54 self.tableview.tableFooterView=footerview;

55 //设置控制

56 footerview.controller=self;

57 }

58 #pragma mark- 懒加载

59 -(NSArray \*)tg

60 {

61 if (\_tg==nil) {

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

63 NSArray \*temparray=[NSArray arrayWithContentsOfFile:fullpath];

64

65 NSMutableArray \*arrayM=[NSMutableArray arrayWithCapacity:temparray.count];

66 for (NSDictionary \*dict in temparray) {

67 YYtgModel \*tg=[YYtgModel tgWithDict:dict];

68 [arrayM addObject:tg];

69 }

70 \_tg=arrayM;

71 }

72 return \_tg;

73 }

74

75 #pragma mark- xib创建cell数据处理

76

77 #pragma mark 多少组

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

79 {

80 return 1;

81 }

82

83 #pragma mark多少行

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

85 {

86 return self.tg.count;

87 }

88

89 #pragma mark设置每组每行

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

91 {

92 //1.创建cell

93 YYtgcell \*cell=[YYtgcell tgcellWithTableView:tableView];

94

95 //2.获取当前行的模型,设置cell的数据

96 YYtgModel \*tg=self.tg[indexPath.row];

97 cell.yytg=tg;

98

99 //3.返回cell

100 return cell;

101 }

102

103 #pragma mark- 隐藏状态栏

104 -(BOOL)prefersStatusBarHidden

105 {

106 return YES;

107 }

108

109 @end

[](javascript:void(0);)

2.通过代理完成

当按钮被点击的时候，视图部分本身不干活，而是通知它的代理（控制器）完成接下来的操作。

该部分代码在1的基础上对下面几个文件进行了修改：

视图部分：

YYfooterview.h文件

[](javascript:void(0);)

1 //

2 // YYfooterview.h

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import <UIKit/UIKit.h>

10 @class YYViewController ,YYfooterview;

11 //约定协议

12 @protocol YYfooterviewDelegate <NSObject>

13 -(void)footerviewLoadMore;

14 @end

15

16 @interface YYfooterview : UIView

17

18 //声明一个id类型属性，遵守了协议的“人”即可成为它的代理

19 @property(nonatomic,strong)id<YYfooterviewDelegate> delegate;

20 //@property(nonatomic,strong) YYViewController \*controller;

21 @end

[](javascript:void(0);)

YYfooterview.m文件

[](javascript:void(0);)

1 //

2 // YYfooterview.m

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import "YYfooterview.h"

10 #import "YYViewController.h"

11

12 @interface YYfooterview ()

13 @property (strong, nonatomic) IBOutlet UIActivityIndicatorView \*loadingview;

14 @property (strong, nonatomic) IBOutlet UIButton \*loadbtn;

15

16 @end

17 @implementation YYfooterview

18 - (IBAction)loadbtclick {

19 NSLog(@"按钮被点击了");

20 //隐藏按钮

21 self.loadbtn.hidden=YES;

22 //显示菊花

23 self.loadingview.hidden=NO;

24

25 #warning 模拟发送网络请求, 3秒之后隐藏菊花

26 dispatch\_after(dispatch\_time(DISPATCH\_TIME\_NOW, (int64\_t)(3.0 \* NSEC\_PER\_SEC)), dispatch\_get\_main\_queue(), ^{

27 // 3.调用控制的加载数据方法

28 // [self.controller LoadMore];

29 //通知代理

30 [self.delegate footerviewLoadMore];

31 // 4.隐藏菊花视图

32 self.loadingview.hidden = YES;

33 // 5.显示按钮

34 self.loadbtn.hidden = NO;

35 });

36 }

37

38 @end

[](javascript:void(0);)

主控制器部分

YYViewController.h文件

[](javascript:void(0);)

1 //

2 // YYViewController.h

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import <UIKit/UIKit.h>

10

11 @interface YYViewController : UIViewController

12 //公开接口

13 //- (void)LoadMore;

14 @end

[](javascript:void(0);)

YYViewController.m文件

[](javascript:void(0);)

1 //

2 // YYViewController.m

3 // 02-团购（使用xib和类完成数据展示）

4 //

5 // Created by apple on 14-5-29.

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

7 //

8

9 #import "YYViewController.h"

10 #import "YYtgModel.h"

11 #import "YYtgcell.h"

12 #import "YYfooterview.h"

13

14 @interface YYViewController ()<UITableViewDataSource,UITableViewDelegate,YYfooterviewDelegate>

15 @property (strong, nonatomic) IBOutlet UITableView \*tableview;

16

17 @property(strong,nonatomic)NSMutableArray \*tg;

18 @end

19

20 @implementation YYViewController

21

22 #pragma mark-加载数据方法

23 -(void)footerviewLoadMore

24 {

25 //创建模型

26 YYtgModel \*tgmodel=[[YYtgModel alloc]init];

27 tgmodel.title=@"菜好上桌";

28 tgmodel.icon=@"5ee372ff039073317a49af5442748071";

29 tgmodel.buyCount=@"20";

30 tgmodel.price=@"10000";

31 //将模型添加到数组中

32 [self.tg addObject:tgmodel];

33

34 YYtgModel \*tgmodelq=[[YYtgModel alloc]init];

35 tgmodelq.title=@"菜好上桌1";

36 tgmodelq.icon=@"5ee372ff039073317a49af5442748071";

37 tgmodelq.buyCount=@"20";

38 tgmodelq.price=@"10000";

39

40 [self.tg addObject:tgmodelq];

41 //刷新表格

42 [self.tableview reloadData];

43 }

44 //-(void)LoadMore

45 //{

46 // //创建模型

47 // YYtgModel \*tgmodel=[[YYtgModel alloc]init];

48 // tgmodel.title=@"菜好上桌";

49 // tgmodel.icon=@"5ee372ff039073317a49af5442748071";

50 // tgmodel.buyCount=@"20";

51 // tgmodel.price=@"10000";

52 // //将模型添加到数组中

53 // [self.tg addObject:tgmodel];

54 //

55 // YYtgModel \*tgmodelq=[[YYtgModel alloc]init];

56 // tgmodelq.title=@"菜好上桌1";

57 // tgmodelq.icon=@"5ee372ff039073317a49af5442748071";

58 // tgmodelq.buyCount=@"20";

59 // tgmodelq.price=@"10000";

60 //

61 // [self.tg addObject:tgmodelq];

62 // //刷新表格

63 // [self.tableview reloadData];

64 //}

65

66 - (void)viewDidLoad

67 {

68 [super viewDidLoad];

69 self.tableview.rowHeight=80.f;

70

71 //加载底部视图

72 //从xib中获取数据

73 UINib \*nib=[UINib nibWithNibName:@"YYfooterview" bundle:nil];

74 YYfooterview \*footerview=[[nib instantiateWithOwner:nil options:nil] firstObject];

75 self.tableview.tableFooterView=footerview;

76 //设置控制

77 // footerview.controller=self;

78 //设置代理

79 footerview.delegate=self;

80 }

81 #pragma mark- 懒加载

82 -(NSArray \*)tg

83 {

84 if (\_tg==nil) {

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

86 NSArray \*temparray=[NSArray arrayWithContentsOfFile:fullpath];

87

88 NSMutableArray \*arrayM=[NSMutableArray arrayWithCapacity:temparray.count];

89 for (NSDictionary \*dict in temparray) {

90 YYtgModel \*tg=[YYtgModel tgWithDict:dict];

91 [arrayM addObject:tg];

92 }

93 \_tg=arrayM;

94 }

95 return \_tg;

96 }

97

98 #pragma mark- xib创建cell数据处理

99

100 #pragma mark 多少组

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

102 {

103 return 1;

104 }

105

106 #pragma mark多少行

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

108 {

109 return self.tg.count;

110 }

111

112 #pragma mark设置每组每行

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

114 {

115 //1.创建cell

116 YYtgcell \*cell=[YYtgcell tgcellWithTableView:tableView];

117

118 //2.获取当前行的模型,设置cell的数据

119 YYtgModel \*tg=self.tg[indexPath.row];

120 cell.yytg=tg;

121

122 //3.返回cell

123 return cell;

124 }

125

126 #pragma mark- 隐藏状态栏

127 -(BOOL)prefersStatusBarHidden

128 {

129 return YES;

130 }

131

132 @end