**单组数据展示**

**Shop.h**

#import <Foundation/Foundation.h>

@interface Shop : NSObject

//图片

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

//名称

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

//描述

@property(nonatomic,copy)NSString \*desc;

+(id)shopWithName:(NSString \*)name icon:(NSString \*)icon desc:(NSString \*)desc;

@end

**Shop.m**

#import "Shop.h"

@implementation Shop

+(id)shopWithName:(NSString \*)name icon:(NSString \*)icon desc:(NSString \*)desc

{

Shop \*shop = [[Shop alloc] init];

shop.icon = icon;

shop.name = name;

shop.desc = desc;

return shop;

}

@end

**ViewController.m**

#import "ViewController.h"

#import "Shop.h"

@interface ViewController ()<UITableViewDataSource,UITableViewDelegate>

{

NSMutableArray \*\_shops;

}

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

//创建数据

\_shops = [NSMutableArray array];

//创建数据

Shop \*shop1 = [Shop shopWithName:@"图书/音像" icon:@"010.png" desc:@"小说、情感"];

Shop \*shop2 = [Shop shopWithName:@"母婴用品" icon:@"011.png" desc:@"奶粉、童装"];

Shop \*shop3 = [Shop shopWithName:@"玩具" icon:@"012.png" desc:@"小汽车、洋娃娃"];

[\_shops addObjectsFromArray:@[shop1,shop2,shop3]];

//添加其他假数据

for (int i = 0; i<20; i++) {

NSString \*name = [NSString stringWithFormat:@"随机产品-%d",i];

NSString \*desc = [NSString stringWithFormat:@"%@-好好好",name];

NSString \*icon = [NSString stringWithFormat:@"01%d.png",i % 9];

Shop \*shop = [Shop shopWithName:name icon:icon desc:desc];

[\_shops addObject:shop];

}

}

#pragma mark 一共1组

//不实现此方法，默认就认为是1组

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

//{

// return 1;

//}

#pragma mark 这一组里面有多少行

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

{

return \_shops.count;

}

#pragma mark 返回第indexPath这行对应的内容

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

{

UITableViewCell \*cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle reuseIdentifier:nil];

//取出这行对应的产品

Shop \*shop = \_shops[indexPath.row];

//产品名称

cell.textLabel.text = shop.name;

//产品描述

cell.detailTextLabel.text = shop.desc;

//产品图片

cell.imageView.image = [UIImage imageNamed:shop.icon];

//设置右边的小图标

cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;

return cell;

}

#pragma mark - 代理方法

#pragma mark 返回indexPath这行cell的高度

-(CGFloat)tableView:(UITableView \*)tableView heightForRowAtIndexPath:(NSIndexPath \*)indexPath

{

return 70;

}

@end

**性能优化**

**Viewcontroller.m**

#import "ViewController.h"

@interface ViewController ()<UITableViewDataSource,UITableViewDelegate>

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

UITableView \*tableView = [[UITableView alloc] initWithFrame:self.view.bounds style:UITableViewStylePlain];

tableView.dataSource = self;

tableView.delegate = self;

[self.view addSubview:tableView];

}

#pragma mark - 数据源方法

#pragma mark 行数

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

{

return 300;

}

#pragma mark 返回每一行的cell

#pragma mark 每当有一个cell进入视野范围内就会调用

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

{

//局部变量只会分配一次存储空间

static NSString \*ID = @"c1";

//NSLog(@"cellForRow---%ld",indexPath.row);

//1、从缓存池中取出可循环利用的cell

UITableViewCell \*cell = [tableView dequeueReusableCellWithIdentifier:ID];

//2、如果缓存池中没有可循环利用的cell

if (cell == nil) {

cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuseIdentifier:ID];

}

cell.textLabel.text = [NSString stringWithFormat:@"第%ld行数据",indexPath.row];

NSLog( @"%p - %ld",cell,indexPath.row);

return cell;

}

#pragma mark - 代理方法

-(CGFloat)tableView:(UITableView \*)tableView heightForRowAtIndexPath:(NSIndexPath \*)indexPath

{

return 100;

}

@end

**性能优化的应用**

**Shop.h**

#import <Foundation/Foundation.h>

@interface Shop : NSObject

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

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

@property(nonatomic,copy)NSString \*desc;

-(id)initWithDict:(NSDictionary \*)dict;

+(id)shopWithDict:(NSDictionary \*)dict;

@end

**Shop.m**

#import "Shop.h"

@implementation Shop

-(id)initWithDict:(NSDictionary \*)dict

{

if (self = [super init]) {

//将字典中的键值对转换成模型属性

self.name = dict[@"name"];

self.icon = dict[@"icon"];

self.desc = dict[@"desc"];

}

return self;

}

+(id)shopWithDict:(NSDictionary \*)dict

{

Shop \*s = [[self alloc] initWithDict:dict];

return s;

}

@end

**ViewController.h**

#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

@property (weak, nonatomic) IBOutlet UILabel \*titleLabel;

@property (weak, nonatomic) IBOutlet UIBarButtonItem \*removeItem;

@property (weak, nonatomic) IBOutlet UITableView \*tableView;

-(IBAction)remove;

@end

**ViewController.m**

#import "ViewController.h"

#import "Shop.h"

@interface ViewController ()<UITableViewDataSource,UITableViewDelegate>

{

NSMutableArray \*\_shops;

NSMutableArray \*\_deleteShops; //存放即将被删除的模型数据

}

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

//1、加载Plist，array中装的都是NSDictionary对象

NSString \*path = [[NSBundle mainBundle] pathForResource:@"shops" ofType:@"plist"];

NSArray \*array = [NSArray arrayWithContentsOfFile:path];

//2、将array中的字典对象转成Shop模型对象

\_shops = [NSMutableArray array];

for (NSDictionary \*dict in array) {

//2.1 创建模型

Shop \*s = [[Shop alloc] initWithDict:dict];

// Shop \*s = [[Shop alloc] init];

//2.2 将字典的键值对（key-value）转成对应的模型属性

// s.name = dict[@"name"];

// s.icon = dict[@"icon"];

// s.desc = dict[@"desc"];

//2.3 将模型对象放进数组中

[\_shops addObject:s];

}

\_deleteShops = [NSMutableArray array];

}

-(void)remove

{

//1、删除模型数据

[\_shops removeObjectsInArray:\_deleteShops];

//2、刷新表格

[self.tableView reloadData];

//3、更改标题

\_titleLabel.text = @"淘宝";

//4、清空\_deleteShops(清空已经删除的数据)

[\_deleteShops removeAllObjects];

//5、让删除item不能被点击

\_removeItem.enabled = NO;

}

#pragma mark - 数据源方法

#pragma mark 多少行数据

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

{

return \_shops.count;

}

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

{

//1、定义一个标识

static NSString \*ID = @"cell";

//2、去缓存池中取出可循环利用的cell

UITableViewCell \*cell = [tableView dequeueReusableCellWithIdentifier:ID];

//3、如果缓存中没有可循环利用的cell

if (cell == nil) {

cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleSubtitle reuseIdentifier:ID];

}

//4、显示商品数据

// 4.0 取出indexPath这行对应的商品模型对象

Shop \*s = \_shops[indexPath.row];

//4.1 商品名称

cell.textLabel.text = s.name;

//4.2 商品描述

cell.detailTextLabel.text = s.desc;

//4.3 商品图片

cell.imageView.image = [UIImage imageNamed:s.icon];

//4.4 检测打钩状态

if ([\_deleteShops containsObject:s]) { //需要打钩

cell.accessoryType = UITableViewCellAccessoryCheckmark;

}else{ //不需要打钩

cell.accessoryType = UITableViewCellAccessoryNone;

}

return cell;

}

#pragma mark - 代理方法

#pragma mark 每一行的高度

-(CGFloat)tableView:(UITableView \*)tableView heightForRowAtIndexPath:(NSIndexPath \*)indexPath

{

return 80;

}

#pragma mark 监听cell的点击

#pragma mark 选中了某一行就会调用

-(void)tableView:(UITableView \*)tableView didSelectRowAtIndexPath:(NSIndexPath \*)indexPath

{

//每个cell都有选中状态（selected）

NSLog(@"---select---%ld",indexPath.row);

//0、取消选中这一行（去掉cell默认的蓝色背景）

[tableView deselectRowAtIndexPath:indexPath animated:YES];

//1、取出indexPath这行对应的cell

// UITableViewCell \*selectedCell = [tableView cellForRowAtIndexPath:indexPath];

//

// selectedCell.accessoryType = UITableViewCellAccessoryCheckmark;

//1、控制当前行的数据是否需要选中

Shop \*s = \_shops[indexPath.row];

if ([\_deleteShops containsObject:s]) { //之前是选中，现在取消选中

[\_deleteShops removeObject:s];

}else{ //之前没有选中，现在选中

[\_deleteShops addObject:s];

}

//2、刷新表格

//[tableView reloadData];

[tableView reloadRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationMiddle];

// for (Shop \*shop in \_deleteShops) {

// NSLog(@"%@",shop.name);

// }

//3、显示标题

if (\_deleteShops.count == 0) {

\_titleLabel.text = @"淘宝";

\_removeItem.enabled = NO;

}else{

\_titleLabel.text = [NSString stringWithFormat:@"淘宝（%ld）",\_deleteShops.count];

\_removeItem.enabled = YES;

}

}

//#pragma mark 取消选中某一行会调用

//-(void)tableView:(UITableView \*)tableView didDeselectRowAtIndexPath:(NSIndexPath \*)indexPath

//{

// NSLog(@"---deselect---%ld",indexPath.row);

//}

@end