**多组数据展示--多维数组**

**ViewController.m**

#import "ViewController.h"

@interface ViewController ()<UITableViewDataSource>

{

//所有的城市

NSArray \*\_allCities;

}

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

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

tableView.dataSource = self;

[self.view addSubview:tableView];

\_allCities = @[

@[@"广州",@"深圳",@"梅州",@"东莞"],

@[@"长沙",@"益阳",@"岳阳"],

@[@"武汉",@"黄冈"],

@[@"桂林"]

];

}

#pragma mark - 数据源方法

#pragma mark 一共有多少组

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

{

return \_allCities.count;

}

#pragma mark 第section组有多少行

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

{

//取得第section组的所有城市

NSArray \*sectionCities = \_allCities[section];

//第section组城市的个数

return sectionCities.count;

}

#pragma mark 返回每一行显示的内容（每行显示怎样的cell）

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

{

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

//indexPath标识唯一的一行，第section组的第row行

cell.textLabel.text = \_allCities[indexPath.section][indexPath.row];

return cell;

}

#pragma mark 第section组显示的头部标题

-(NSString \*)tableView:(UITableView \*)tableView titleForHeaderInSection:(NSInteger)section

{

if (section == 0) return @"广东";

if (section == 1) return @"湖南";

if (section == 2) return @"湖北";

if (section == 3) return @"广西";

return nil;

}

#pragma mark 第section组显示的尾部标题

-(NSString \*)tableView:(UITableView \*)tableView titleForFooterInSection:(NSInteger)section

{

if (section == 0) return @"广东省会是广州";

if (section == 1) return @"湖南省会是长沙";

if (section == 2) return @"湖北省会是武汉";

if (section == 3) return @"广西省会是南宁";

return nil;

}

@end

**多组数据展示—字典**

**ViewController.m**

#import "ViewController.h"

//省份字典中用到的key

#define kHeader @"header" //头部标题对应的key

#define kFooter @"footer" //尾部标题对应的key

#define kCities @"cities" //城市数组对应的key

@interface ViewController ()<UITableViewDataSource>

{

//所有的省份

NSArray \*\_allProvinces;

}

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

//1、添加tableView

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

tableView.dataSource = self;

[self.view addSubview:tableView];

//2、初始化数据

\_allProvinces = @[

@{

kHeader:@"广东",

kFooter:@"广东的省会是广州",

kCities:@[@"广州",@"深圳",@"梅州",@"东莞"]

},

@{

kHeader:@"湖南",

kFooter:@"湖南的省会是长沙",

kCities:@[@"长沙",@"益阳",@"岳阳"],

},

@{

kHeader:@"湖北",

kFooter:@"湖北的省会是武汉",

kCities:@[@"武汉",@"黄冈"]

},

@{

kHeader:@"广西",

kFooter:@"广西的省会是南宁",

kCities: @[@"桂林"]

}

];

}

#pragma mark - 数据源方法

#pragma mark 一共有多少组

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

{

return \_allProvinces.count;

}

#pragma mark 第section组有多少行

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

{

//1、取得第section组的省份

NSDictionary \*province = \_allProvinces[section];

//2、取得省份里面的城市数组

NSArray \*cities = province[kCities];

return cities.count;

}

#pragma mark 返回每一行显示的内容（每行显示怎样的cell）

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

{

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

//indexPath标识唯一的一行，第section组的第row行

//取出第section组的省份中城市数组里面第row行的数据

NSDictionary \*province = \_allProvinces[indexPath.section];

NSArray \*cities = province[kCities];

cell.textLabel.text = cities[indexPath.row];

return cell;

}

#pragma mark 第section组显示的头部标题

-(NSString \*)tableView:(UITableView \*)tableView titleForHeaderInSection:(NSInteger)section

{

return \_allProvinces[section][kHeader];

}

#pragma mark 第section组显示的尾部标题

-(NSString \*)tableView:(UITableView \*)tableView titleForFooterInSection:(NSInteger)section

{

return \_allProvinces[section][kFooter];

}

@end

**多组数据展示—数据模型**

**Province.h**

#import <Foundation/Foundation.h>

@interface Province : NSObject

//UI控件用weak,NSString用copy,其他对象一般用strong

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

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

@property(nonatomic,strong) NSArray \*cities;

+(id)provinceWithHeader:(NSString \*)header footer:(NSString \*)footer cities:(NSArray \*)cities;

@end

**Province.m**

#import "Province.h"

@implementation Province

+(id)provinceWithHeader:(NSString \*)header footer:(NSString \*)footer cities:(NSArray \*)cities

{

Province \*p = [[Province alloc] init];

p.header = header;

p.footer = footer;

p.cities = cities;

return p;

}

@end

**ViewController.m**

#import "ViewController.h"

#import "Province.h"

//省份字典中用到的key

//#define kHeader @"header" //头部标题对应的key

//#define kFooter @"footer" //尾部标题对应的key

//#define kCities @"cities" //城市数组对应的key

@interface ViewController ()<UITableViewDataSource>

{

//所有的省份

NSArray \*\_allProvinces;

}

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

//1、添加tableView

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

tableView.dataSource = self;

[self.view addSubview:tableView];

//2、初始化数据(模型对象，数据模型，Model，仅仅是用来存放数据的)

// Province \*gd = [[Province alloc] init];

// gd.header = @"广东";

// gd.footer = @"广东的省会是广州";

// gd.cities = @[@"广州",@"深圳",@"梅州",@"东莞"];

//

// Province \*hn = [[Province alloc] init];

// hn.header = @"湖南";

// hn.footer = @"湖南的省会是长沙";

// hn.cities = @[@"长沙",@"益阳",@"岳阳"];

//

// Province \*hb = [[Province alloc] init];

// hb.header = @"湖北";

// hb.footer = @"湖北的省会是武汉";

// hb.cities = @[@"武汉",@"黄冈"];

Province \*gd = [Province provinceWithHeader:@"广东" footer:@"广东的省会是广州" cities:@[@"广州",@"深圳",@"梅州",@"东莞"]];

Province \*hn = [Province provinceWithHeader:@"湖南" footer:@"湖南的省会是长沙" cities:@[@"长沙",@"益阳",@"岳阳"]];

\_allProvinces = @[gd,hn];

}

#pragma mark - 数据源方法

#pragma mark 一共有多少组

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

{

return \_allProvinces.count;

}

#pragma mark 第section组有多少行

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

{

//1、取得第section组的省份

Province \*province = \_allProvinces[section];

//2、取得省份里面的城市数组

NSArray \*cities = province.cities;

return cities.count;

}

#pragma mark 返回每一行显示的内容（每行显示怎样的cell）

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

{

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

//indexPath标识唯一的一行，第section组的第row行

//取出第section组的省份中城市数组里面第row行的数据

Province \*province = \_allProvinces[indexPath.section];

NSArray \*cities = province.cities;

cell.textLabel.text = cities[indexPath.row];

return cell;

}

#pragma mark 第section组显示的头部标题

-(NSString \*)tableView:(UITableView \*)tableView titleForHeaderInSection:(NSInteger)section

{

Province \*province = \_allProvinces[section];

return province.header;

//return \_allProvinces[section].header; id类型不用调用点语法

}

#pragma mark 第section组显示的尾部标题

-(NSString \*)tableView:(UITableView \*)tableView titleForFooterInSection:(NSInteger)section

{

Province \*province = \_allProvinces[section];

return province.footer;

}

@end

**单组数据展示**

**ViewController.m**

#import "ViewController.h"

@interface ViewController ()<UITableViewDataSource,UITableViewDelegate>

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

}

#pragma mark 一共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 9;

}

#pragma mark 返回第indexPath这行对应的内容

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

{

/\*

Default:不显示detailTextLabet

Value1:在右边显示detailTextLabet

Value2:不显示图片，显示detailTextLabet

Subtitle:在底部显示detailTextLabet

\*/

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

cell.textLabel.text = [NSString stringWithFormat:@"产品-%ld",indexPath.row];

NSString \*imgName = [NSString stringWithFormat:@"01%ld.png",indexPath.row];

cell.imageView.image = [UIImage imageNamed:imgName];

//设置详情文字

cell.detailTextLabel.text = [NSString stringWithFormat:@"产品%ld非常好",indexPath.row];

//设置右边的小图标

//cell.accessoryType = UITableViewCellAccessoryDisclosureIndicator;

//设置最右边显示什么控件

cell.accessoryView = [UIButton buttonWithType:UIButtonTypeContactAdd];

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