**8、美化**

**（1）AppDelegate**

import UIKit

@UIApplicationMain

class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?

func application(\_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {

//设置导航条背景颜色

UINavigationBar.appearance().barTintColor = UIColor(red: 242/255, green: 116/255, blue: 119/255, alpha: 1)

//设置前景色

UINavigationBar.appearance().tintColor = UIColor.white

//设置字体

if let barFont = UIFont(name: "Avenir-Light", size: 24){

UINavigationBar.appearance().titleTextAttributes = [

//设置字体名称

NSFontAttributeName: barFont,

//设置前景色

NSForegroundColorAttributeName: UIColor.white

]

}

return true

}

}

**（2）Area**

import Foundation

//建立一个结构体模型

struct Area {

var name : String

var province : String

var part : String

var image : String

var isVisited : Bool

}

**（3）AreaTableViewController**

import UIKit

class AreaTableViewController: UITableViewController {

var areas = [数组内容省略]

]

override func viewDidLoad() {

super.viewDidLoad()

//定义一个估计行高

tableView.estimatedRowHeight = 80

//行高自适应

tableView.rowHeight = UITableViewAutomaticDimension

//自定义详情页导航条格式

self.navigationItem.backBarButtonItem = UIBarButtonItem(title: "", style: .plain, target: nil, action: nil)

}

// MARK: - Table view dalegate

override func tableView(\_ tableView: UITableView, editActionsForRowAt indexPath: IndexPath) -> [UITableViewRowAction]? {

编码省略

}

//下面一行是为了分隔提示

// MARK: - Table view data source

override func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return areas.count

}

override func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

代码省略

}

//插入或者删除指定行

override func tableView(\_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {

//删除行

if editingStyle == .delete{

//从数据源中删除相应的行

self.areas.remove(at: indexPath.row)

//删除某一行刷新

tableView.deleteRows(at: [indexPath], with: .fade)

//整体刷新

//tableView.reloadData()

}else if editingStyle == .insert{

}

}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

//判断转场的identifier是否与自定义的相等

if segue.identifier == "showAreaDetail"{

//获取转场目标的控制器

let dest = segue.destination as! DetailTableViewController

//设置目标控制器图像的名称，来自于area数组中的某一项，由当前tablevie中的行决定数组下标

dest.area = areas[(tableView.indexPathForSelectedRow?.row)!]

}

}

}

**（4）DetailTableViewController**

import UIKit

class DetailTableViewController: UITableViewController {

@IBOutlet weak var largeImageView: UIImageView!

var area: Area!

override func viewDidLoad() {

super.viewDidLoad()

//设置图片

largeImageView.image = UIImage(named: area.image)

//设置背景颜色稍微偏灰，不透明

tableView.backgroundColor = UIColor(white: 0.98, alpha: 1)

//设置底部为一个大小为0的矩形

tableView.tableFooterView = UIView(frame: CGRect.zero)

//设置分割线颜色

tableView.separatorColor = UIColor(white: 0.9, alpha: 1)

//预计行高

tableView.estimatedRowHeight = 40

//实际行高为自动适应

tableView.rowHeight = UITableViewAutomaticDimension

//设置详情页标题

self.title = area.name

}

// MARK: - Table view data source

override func tableView(\_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return 4

}

override func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

//创建自定义cell

let cell = tableView.dequeueReusableCell(withIdentifier: "DetailCell", for: indexPath) as! DetailTableViewCell

//单元格透明色

cell.backgroundColor = UIColor.clear

//通过switch语句给出四行的显示内容

switch indexPath.row {

case 0:

cell.fieldLabel.text = "地名"

cell.valueLabel.text = area.name

case 1:

cell.fieldLabel.text = "省"

cell.valueLabel.text = area.province

case 2:

cell.fieldLabel.text = "地区"

cell.valueLabel.text = area.part

case 3:

cell.fieldLabel.text = "去过与否"

cell.valueLabel.text = area.isVisited ? "去过":"还没去过"

default:

break

}

return cell

}

}

**9、动画及反场**

**（1）Area**

import Foundation

//建立一个结构体模型

struct Area {

var name : String

var province : String

var part : String

var image : String

var isVisited : Bool

var rating = ""

//保持以前数据的兼容

init(name: String, province: String, part: String, image: String, isVisited: Bool){

self.name = name

self.province = province

self.part = part

self.image = image

self.isVisited = isVisited

}

}

**（2）DetailTableViewController**

import UIKit

class DetailTableViewController: UITableViewController {

@IBAction func close(segue: UIStoryboardSegue){

//反场前取回源控制器

let reviewVC = segue.source as! ReviewViewController

//得到源控制中rating中的值，并判断其中是否有值

if let rating = reviewVC.rating {

//更新模型中的rating值

self.area.rating = rating

//将当前ratingBtn中的图片设置为rating

self.ratingBtn.setImage(UIImage(named: rating), for: .normal)

}

}

**}**

**（3）ReviewViewController**

import UIKit

class ReviewViewController: UIViewController {

var rating: String?

@IBOutlet weak var bgImageView: UIImageView!

@IBOutlet weak var ratingStackView: UIStackView!

//点击评价按钮

@IBAction func ratingTap(\_ sender: UIButton) {

//判断点击的是哪一个按钮，给评价rating赋值

switch sender.tag {

case 100:

rating = "dislike"

case 101:

rating = "general"

case 102:

rating = "good"

default:

break

}

//用代码实现转场

performSegue(withIdentifier: "unwindToDetailView", sender: self)

}

override func viewDidLoad() {

super.viewDidLoad()

//指定一个特效类型

let effect = UIBlurEffect(style: .light)

//创建一个特效视图

let effeciView = UIVisualEffectView(effect: effect)

//设定特效视图的大小

effeciView.frame = view.frame

//把特效视图叠加在原视图之上

bgImageView.addSubview(effeciView)

//创建初始为0，加载视图的时候不显示

//ratingStackView.transform = CGAffineTransform(scaleX: 0, y: 0)

//设置起始位置、大小

let startPos = CGAffineTransform(translationX: 0, y: 500)

let startScale = CGAffineTransform(scaleX: 0, y: 0)

//将两个transform组合一起,注意一定要用放缩组合位置

ratingStackView.transform = startScale.concatenating(startPos)

}

//试图加载完毕后调用

override func viewDidAppear(\_ animated: Bool) {

/\*UIView.animate(withDuration: 0.5) {

//将ratingStackView的transform设置成原始大小

self.ratingStackView.transform = CGAffineTransform.identity

}\*/

//Spring动画，有震动的效果

//delay延迟时间为0，Damping震荡时间为0.3，initialSpringVelocity初始速度为0.5，options选项为空数组

UIView.animate(withDuration: 1, delay: 0, usingSpringWithDamping: 0.3, initialSpringVelocity: 0.5, options: [], animations: {

//设置动画结束位置和大小

let endPos = CGAffineTransform(translationX: 0, y: 0)

let endScale = CGAffineTransform.identity

//self.ratingStackView.transform = CGAffineTransform.identity

//将两个变化组合到一起,注意一定要用放缩组合位置

self.ratingStackView.transform = endScale.concatenating(endPos)

}, completion: nil)

}

}