MealTableViewController.swift

//

// Created by Jane Appleseed on 11/15/16.

// Copyright © 2016 Apple Inc. All rights reserved.

//

import UIKit

import os.log

class MealTableViewController: UITableViewController {

//MARK: Properties

var meals = [Meal]()

override func viewDidLoad() {

super.viewDidLoad()

// Use the edit button item provided by the table view controller.

navigationItem.leftBarButtonItem = editButtonItem

// Load any saved meals, otherwise load sample data.

if let savedMeals = loadMeals() {

meals += savedMeals

}

else {

// Load the sample data.

loadSampleMeals()

}

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

//MARK: - Table view data source

override func numberOfSections(in tableView: UITableView) -> Int {

return 1

}

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

return meals.count

}

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

// Table view cells are reused and should be dequeued using a cell identifier.

let cellIdentifier = "MealTableViewCell"

guard let cell = tableView.dequeueReusableCell(withIdentifier: cellIdentifier, for: indexPath) as? MealTableViewCell else {

fatalError("The dequeued cell is not an instance of MealTableViewCell.")

}

// Fetches the appropriate meal for the data source layout.

let meal = meals[indexPath.row]

cell.nameLabel.text = meal.name

cell.photoImageView.image = meal.photo

cell.ratingControl.rating = meal.rating

return cell

}

// Override to support conditional editing of the table view.

override func tableView(\_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {

// Return false if you do not want the specified item to be editable.

return true

}

// Override to support editing the table view.

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

if editingStyle == .delete {

// Delete the row from the data source

meals.remove(at: indexPath.row)

saveMeals()

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

} else if editingStyle == .insert {

// Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view

}

}

/\*

// Override to support rearranging the table view.

override func tableView(\_ tableView: UITableView, moveRowAt fromIndexPath: IndexPath, to: IndexPath) {

}

\*/

/\*

// Override to support conditional rearranging of the table view.

override func tableView(\_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool {

// Return false if you do not want the item to be re-orderable.

return true

}

\*/

//MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation

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

super.prepare(for: segue, sender: sender)

switch(segue.identifier ?? "") {

case "AddItem":

os\_log("Adding a new meal.", log: OSLog.default, type: .debug)

case "ShowDetail":

guard let mealDetailViewController = segue.destination as? MealViewController else {

fatalError("Unexpected destination: \(segue.destination)")

}

guard let selectedMealCell = sender as? MealTableViewCell else {

fatalError("Unexpected sender: \(sender)")

}

guard let indexPath = tableView.indexPath(for: selectedMealCell) else {

fatalError("The selected cell is not being displayed by the table")

}

let selectedMeal = meals[indexPath.row]

mealDetailViewController.meal = selectedMeal

default:

fatalError("Unexpected Segue Identifier; \(segue.identifier)")

}

}

//MARK: Actions

@IBAction func unwindToMealList(sender: UIStoryboardSegue) {

if let sourceViewController = sender.source as? MealViewController, let meal = sourceViewController.meal {

if let selectedIndexPath = tableView.indexPathForSelectedRow {

// Update an existing meal.

meals[selectedIndexPath.row] = meal

tableView.reloadRows(at: [selectedIndexPath], with: .none)

}

else {

// Add a new meal.

let newIndexPath = IndexPath(row: meals.count, section: 0)

meals.append(meal)

tableView.insertRows(at: [newIndexPath], with: .automatic)

}

// Save the meals.

saveMeals()

}

}

//MARK: Private Methods

private func loadSampleMeals() {

let photo1 = UIImage(named: "meal1")

let photo2 = UIImage(named: "meal2")

let photo3 = UIImage(named: "meal3")

guard let meal1 = Meal(name: "Caprese Salad", photo: photo1, rating: 4) else {

fatalError("Unable to instantiate meal1")

}

guard let meal2 = Meal(name: "Chicken and Potatoes", photo: photo2, rating: 5) else {

fatalError("Unable to instantiate meal2")

}

guard let meal3 = Meal(name: "Pasta with Meatballs", photo: photo3, rating: 3) else {

fatalError("Unable to instantiate meal2")

}

meals += [meal1, meal2, meal3]

}

private func saveMeals() {

let isSuccessfulSave = NSKeyedArchiver.archiveRootObject(meals, toFile: Meal.ArchiveURL.path)

if isSuccessfulSave {

os\_log("Meals successfully saved.", log: OSLog.default, type: .debug)

} else {

os\_log("Failed to save meals...", log: OSLog.default, type: .error)

}

}

private func loadMeals() -> [Meal]? {

return NSKeyedUnarchiver.unarchiveObject(withFile: Meal.ArchiveURL.path) as? [Meal]

}

}