//

// ViewController.swift

// FillTableViewWithAPIData

//

// Created by user226203 on 9/22/22.

//

import UIKit

class ViewController: UIViewController {

private let tableView = UITableView()

private var safeArea: UILayoutGuide!

private let names = ["Ramu", "Shyamu", "Mohan", "Maadhava"]

private let profession = ["Student", "iOS developer", "Hard worker", "God"]

private var dataArray = [AmiiboModel]()

override func viewDidLoad() {

super.viewDidLoad()

safeArea = view.layoutMarginsGuide

//MARK: - Get API data here

NetworkHandler.shared.getAPIData() {

[weak self] dataModel in

DispatchQueue.main.async {

self?.dataArray = dataModel

self?.tableView.reloadData()

}

}

//tableView.register(UITableViewCell.self, forCellReuseIdentifier: "cell")

tableView.register(CustomTableViewCell.self, forCellReuseIdentifier: "cell")

tableView.dataSource = self

tableView.delegate = self

setUpTableViw()

}

func setUpTableViw() {

tableView.translatesAutoresizingMaskIntoConstraints = false

view.addSubview(tableView)

tableView.leadingAnchor.constraint(equalTo: safeArea.leadingAnchor).isActive = true

tableView.trailingAnchor.constraint(equalTo: safeArea.trailingAnchor).isActive = true

tableView.topAnchor.constraint(equalTo: safeArea.topAnchor).isActive = true

tableView.bottomAnchor.constraint(equalTo: safeArea.bottomAnchor).isActive = true

}

}

extension ViewController: UITableViewDataSource {

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

return dataArray.count

}

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

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

/\* let nameObject = names[indexPath.row]

let professionObject = profession[indexPath.row]

cell.title.text = nameObject

cell.subTitle.text = professionObject\*/

let arrayData = dataArray[indexPath.row]

cell.title.text = arrayData.amiiboSeries

cell.subTitle.text = arrayData.character

return cell

}

}

extension ViewController: UITableViewDelegate {

func tableView(\_ tableView: UITableView, leadingSwipeActionsConfigurationForRowAt indexPath: IndexPath) -> UISwipeActionsConfiguration? {

let action = UIContextualAction(style: .destructive, title: "Delete") {action, view, completionHandler in

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

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

completionHandler(true)

}

return UISwipeActionsConfiguration(actions: [action])

}

func tableView(\_ tableView: UITableView, trailingSwipeActionsConfigurationForRowAt indexPath: IndexPath) -> UISwipeActionsConfiguration? {

let trailingAction = UIContextualAction(style: .destructive, title: "Trailing Delete") {

action, view, completionHandler in

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

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

}

return UISwipeActionsConfiguration(actions: [trailingAction])

}

}