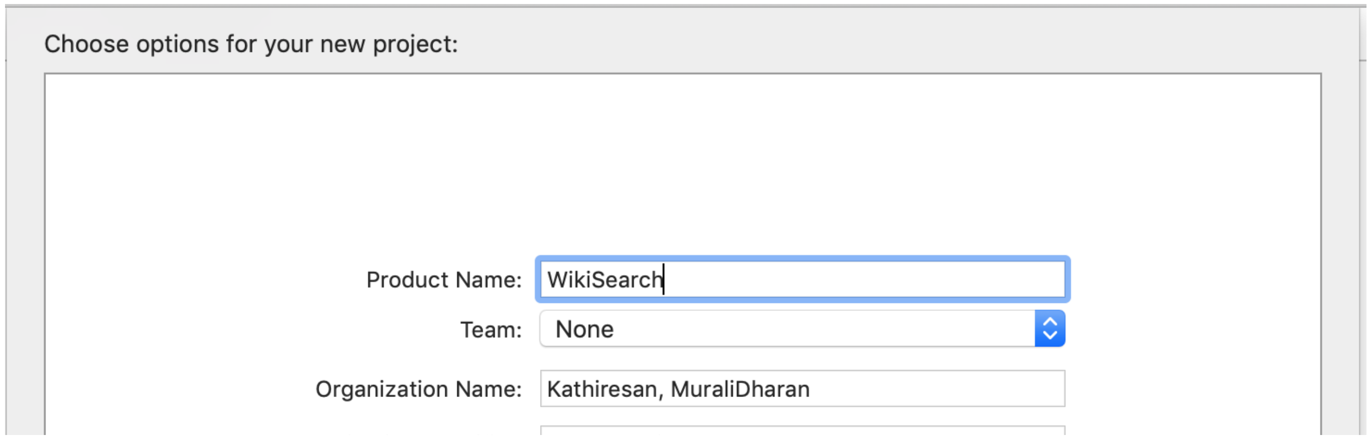
This article helps to design a UITableViewController with a SearchBar and populates results whenever we start typing and follows opening a SafariWebView on tapping on the search results.

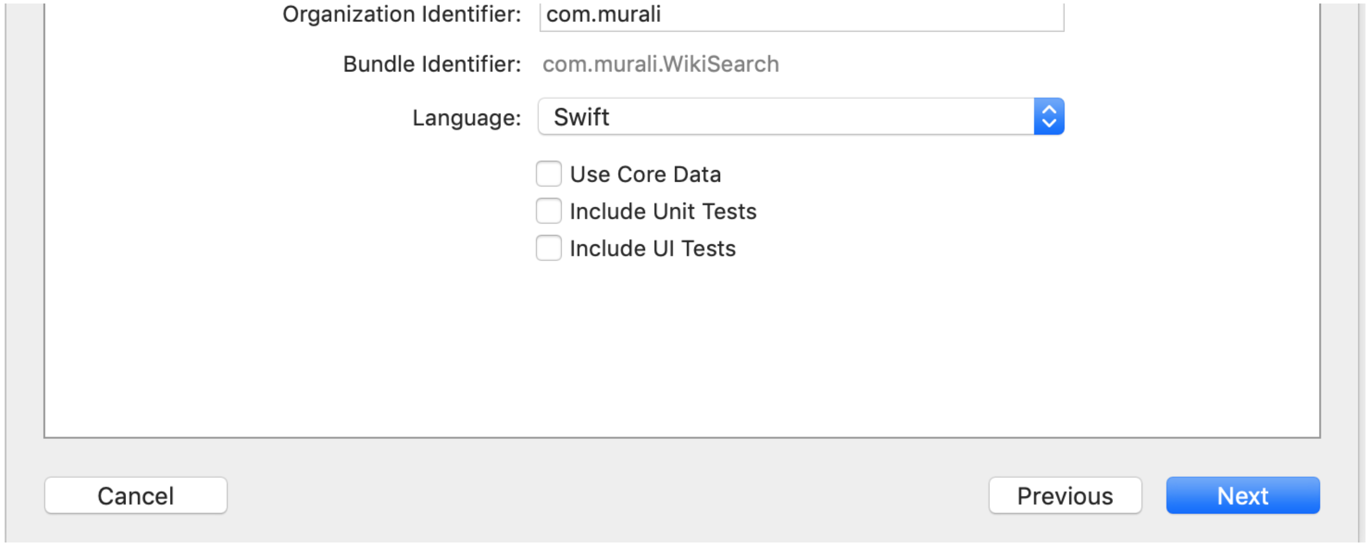
Also, this article helps to understand how to use **Alamofire** with **SwiftyJSON** to hit an API and fetch some results, parse them and show the results in an UITableViewController.

1. Create a new project in Xcode



Choose options for your new project:

Product Name:	WikiSearch
Team:	None
Organization Name:	Kathiresan, MuraliDharan



Organization Identifier: com.murali

Bundle Identifier: com.murali.WikiSearch

Language: Swift

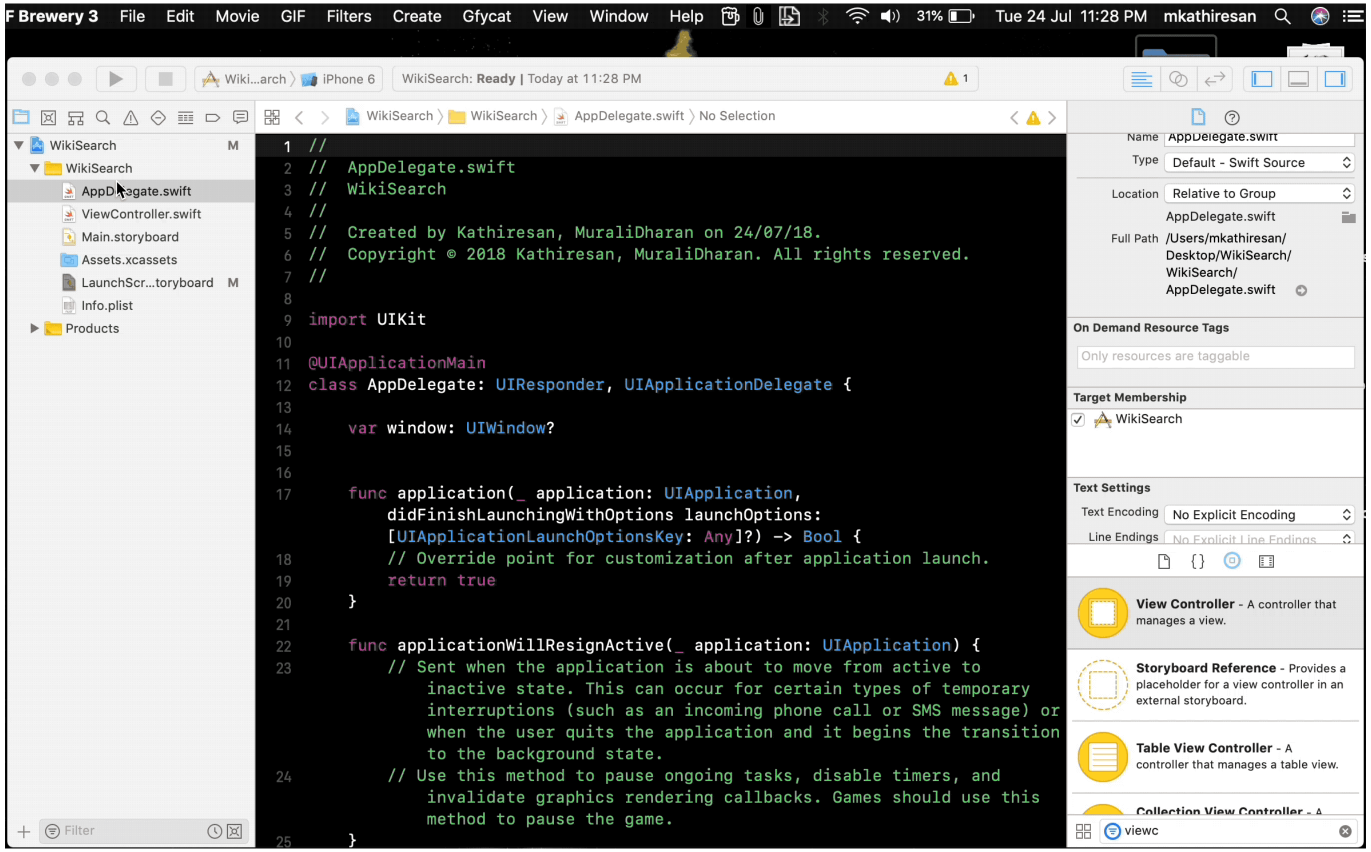
- [ ] Use Core Data
- [ ] Include Unit Tests
- [ ] Include UI Tests

2. Go to terminal and enter the project directory path and type `pod init` this creates `podfile` in our directory and then open this podfile from the project directory and following pods into it

```
platform :ios, '9.0'

target 'WikiSearch' do
     use_frameworks!

  pod 'Alamofire', '~> 4.7'
  pod 'SwiftyJSON', '~> 4.0'

end
```

then run `pod install` in the terminal, this helps us to install these dependencies and resulting in the creation `.xcworkspace` file, close the project and open the *WikiSearch.xcworkspace* file. If you are new to cocoapods, please have a look here

3. Add a UITableViewController with NavigationController in the `Main.storyboard`


F Brewery 3 File Edit Movie GIF Filters Create Gfycat View Window Help
31% Tue 24 Jul 11:28 PM mkathiresan
Wiki...arch iPhone 6
WikiSearch: Ready | Today at 11:28 PM
WikiSearch M
WikiSearch
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScr...toryboard M
Info.plist
Products
Filter
WikiSearch 〉 WikiSearch 〉 AppDelegate.swift 〉 No Selection
viewc

```
//
//  AppDelegate.swift
//  WikiSearch
//
//  Created by Kathiresan, MuraliDharan on 24/07/18.
//  Copyright © 2018 Kathiresan, MuraliDharan. All rights reserved.
//

import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions:
        [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        return true
    }

    func applicationWillResignActive(_ application: UIApplication) {
        // Sent when the application is about to move from active to
            inactive state. This can occur for certain types of temporary
            interruptions (such as an incoming phone call or SMS message) or
            when the user quits the application and it begins the transition
            to the background state.
        // Use this method to pause ongoing tasks, disable timers, and
            invalidate graphics rendering callbacks. Games should use this
            method to pause the game.
    }
```

Name AppDelegate.swift
Type Default - Swift Source
Location Relative to Group
AppDelegate.swift
Full Path /Users/mkathiresan/Desktop/WikiSearch/WikiSearch/AppDelegate.swift

On Demand Resource Tags
Only resources are taggable

Target Membership
✓ WikiSearch

Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings

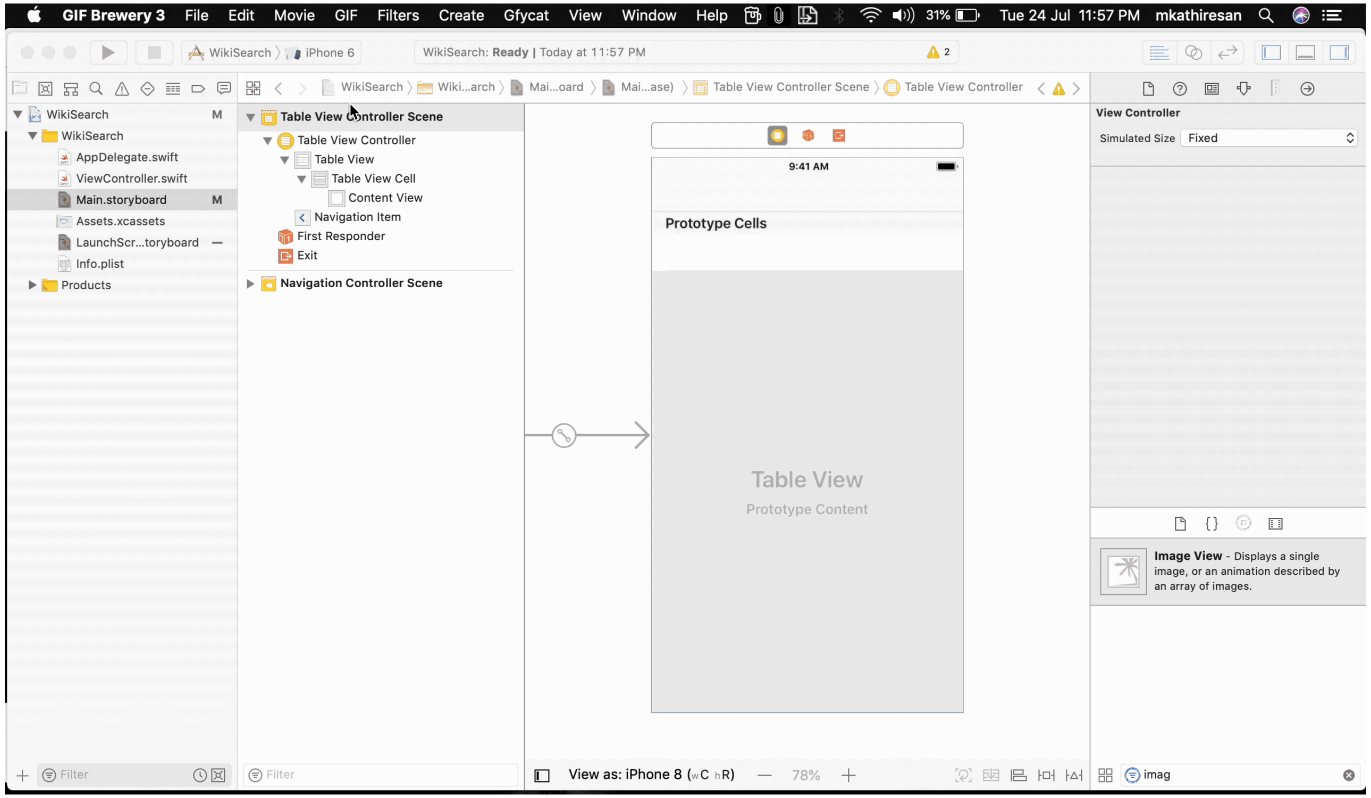
View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

Table View Controller - A controller that manages a table view.

Collection View Controller - A

4. I am going to create an *ImageView*, *TitleLabel* and *DescriptionLabel*, so there will be three UIElements in the TableView cell.


GIF Brewery 3 File Edit Movie GIF Filters Create Gfycat View Window Help
31% Tue 24 Jul 11:57 PM mkathiresan
WikiSearch iPhone 6
WikiSearch: Ready | Today at 11:57 PM
WikiSearch Wiki...arch Mai...oard Mai...ase) Table View Controller Scene Table View Controller
WikiSearch M
WikiSearch
AppDelegate.swift
ViewController.swift
Main.storyboard M
Assets.xcassets
LaunchScr...toryboard
Info.plist
Products
Table View Controller Scene
Table View Controller
Table View
Table View Cell
Content View
Navigation Item
First Responder
Exit
Navigation Controller Scene
9:41 AM
Prototype Cells
Table View
Prototype Content
View Controller
Simulated Size Fixed
Image View - Displays a single image, or an animation described by an array of images.
Filter
View as: iPhone 8 (wC hR)
78%
imag

5. Add a new class `CustomTableViewCell` sub-classing UITableViewCell to the project and add the referencing layouts from the table view cell.

```swift
import UIKit

class CustomTableViewCell: UITableViewCell {

    @IBOutlet weak var wikiImageView: UIImageView!

    @IBOutlet weak var titleLabel: UILabel!

    @IBOutlet weak var descriptionLabel: UILabel!

    override func awakeFromNib() {
        super.awakeFromNib()
    }

    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)
    }
}
```

CustomTableViewCell.swift hosted with ❤ by GitHub view raw

6. Now, add a new class `SearchResultsTableViewController` sub-classing `UITableViewController` and create a UISearchController property in it.

```
private let searchController =
UISearchController(searchResultsController: nil)
```

Let us customise this searchBar and background view for the tableview like below:

```
private func setupSearchBar() {
    searchController.searchBar.delegate = self
    searchController.dimsBackgroundDuringPresentation = false
    searchController.hidesNavigationBarDuringPresentation = false
    searchController.searchBar.placeholder = "Search any Topic"
    definesPresentationContext = true
    tableView.tableHeaderView = searchController.searchBar
}
```

and

```
private func setupTableViewBackgroundView() {
   let backgroundViewLabel = UILabel(frame: .zero)
   backgroundViewLabel.textColor = .darkGray
   backgroundViewLabel.numberOfLines = 0
   backgroundViewLabel.text =
          "Oops, /n No results to show! ..."
   tableView.backgroundView = backgroundViewLabel
}
```

add these methods into `viewDidLoad` of the controller.

```
override func viewDidLoad() {
    super.viewDidLoad()
    tableView.tableFooterView = UIView()
    setupTableViewBackgroundView()
```

```
        setupSearchBar()
    }
```

An UIView is added as tableFooterView so that empty cells will not be visible.

Now let us create an APIFetcher as a helper to helps us to fetch content from the API whenever we type something in the search bar with two methods inside it.

```
func search(searchText: String, completionHandler: @escaping
([JSON]?, NetworkError) -> ()) {}

func fetchImage(url: String, completionHandler: @escaping (UIImage?,
NetworkError) -> ()) {}
```

The first method is used to hit the API with desired search text from the searchBar, the Second method is used to fetch an image from the URL received as a search result, the entire helper class will look like below:

```
import Foundation
import SwiftyJSON
```

```swift
import Alamofire

enum NetworkError: Error {
    case failure
    case success
}

class APIRequestFetcher {
    var searchResults = [JSON]()

    func search(searchText: String, completionHandler: @escaping ([JSON]?, NetworkError)
        let urlToSearch = "https://en.wikipedia.org//w/api.php?action=query&format=json&

        Alamofire.request(urlToSearch).responseJSON { response in
            guard let data = response.data else {
                completionHandler(nil, .failure)
                return
            }

            let json = try? JSON(data: data)
            let results = json?["query"]["pages"].arrayValue
            guard let empty = results?.isEmpty, !empty else {
                completionHandler(nil, .failure)
                return
            }

            completionHandler(results, .success)
        }
    }

    func fetchImage(url: String, completionHandler: @escaping (UIImage?, NetworkError) -
        Alamofire.request(url).responseData { responseData in

```

```swift
            guard let imageData = responseData.data else {
                completionHandler(nil, .failure)
                return
            }

            guard let image = UIImage(data: imageData) else {
                completionHandler(nil, .failure)
                return
            }

            completionHandler(image, .success)
        }
    }
}
```

APIRequestFetcher.swift hosted with ❤ by GitHub view raw

I have created an enumeration to pass the network status in the completion handler.

The API used here to hit and get search results is Wikipedia Media API

Now let us make the controller ready for making this API to hit and show the results,

```swift
import UIKit
import SwiftyJSON
import Alamofire
import Alamofire
import SafariServices

final class SearchResultsTableViewController: UITableViewController {

    private var searchResults = [JSON]() {
        didSet {
            tableView.reloadData()
        }
    }

    private let searchController = UISearchController(searchResultsController: nil)
    private let apiFetcher = APIRequestFetcher()
    private var previousRun = Date()
    private let minInterval = 0.05

    override func viewDidLoad() {
        super.viewDidLoad()
        tableView.tableFooterView = UIView()
        setupTableViewBackgroundView()
        setupSearchBar()
    }

    private func setupTableViewBackgroundView() {
        let backgroundViewLabel = UILabel(frame: .zero)
        backgroundViewLabel.textColor = .darkGray
        backgroundViewLabel.numberOfLines = 0
        backgroundViewLabel.text = " Oops, No results to show "
        backgroundViewLabel.textAlignment = NSTextAlignment.center
        backgroundViewLabel.font.withSize(20)
        tableView.backgroundView = backgroundViewLabel
    }

    private func setupSearchBar() {
        searchController.searchBar.delegate = self
        searchController.dimsBackgroundDuringPresentation = false
        searchController.hidesNavigationBarDuringPresentation = false
        searchController.searchBar.placeholder = "Search any Topic"
        definesPresentationContext = true
        tableView.tableHeaderView = searchController.searchBar
    }

    override func numberOfSections(in tableView: UITableView) -> Int {
        return 1
    }

    override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Ir
        return searchResults.count
    }

    override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath
        let cell = tableView.dequeueReusableCell(withIdentifier: "cell",
                                                  for: indexPath) as! CustomTableViewCel

        cell.titleLabel.text = searchResults[indexPath.row]["title"].stringValue

        cell.descriptionLabel.text = searchResults[indexPath.row]["terms"]["descriptior

        if let url = searchResults[indexPath.row]["thumbnail"]["source"].string {
            apiFetcher.fetchImage(url: url, completionHandler: { image, _ in
                cell.wikiImageView.image = image
            })
        }

        return cell
    }

    override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPa

        let title = searchResults[indexPath.row]["title"].stringValue
        guard let url = URL.init(string: "https://en.wikipedia.org/wiki/\(title)")
            else { return }

        let safariVC = SFSafariViewController(url: url)
        present(safariVC, animated: true, completion: nil)
        tableView.deselectRow(at: indexPath, animated: true)
    }

}

extension SearchResultsTableViewController: UISearchBarDelegate {

    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
        searchResults.removeAll()
        guard let textToSearch = searchBar.text, !textToSearch.isEmpty else {
            return
        }

        if Date().timeIntervalSince(previousRun) > minInterval {
            previousRun = Date()
            fetchResults(for: textToSearch)
        }
    }

    func fetchResults(for text: String) {
        print("Text Searched: \(text)")
        apiFetcher.search(searchText: text, completionHandler: {
            [weak self] results, error in
            if case .failure = error {
                return
            }

            guard let results = results, !results.isEmpty else {
                return
            }

            self?.searchResults = results
        })
    }

    func searchBarCancelButtonClicked(_ searchBar: UISearchBar) {
        searchResults.removeAll()
    }

}
```

SearchResultsTableViewController.swift hosted with ❤ by GitHub view raw

The `didSelectRow` method is configured with SafariServices, the idea is whenever we tap on a cell, it will open the respective Wikipedia page in the app itself.

also, there is a delay added to hit the API after the user typed something on the searchBar to avoid multiple calls to the API unnecessarily

Every result from the API will look like:

```
{
  "index": 5,
  "ns": 0,
  "pageid": 1389932,
  "terms": {
    "description": [
      "Indian cricket player"
    ]
  },
  "thumbnail": {
    "height": 50,
    "source":
"https://upload.wikimedia.org/wikipedia/commons/thumb/6/6c/Murali_ka
rtik_bowling.jpg/25px-Murali_kartik_bowling.jpg",
    "width": 25
  },
  "title": "Murali Kartik"
}
```

among all these, we gonna map the `title` to our title, `description` to our description and `thumbnail source` to our imageView

That's IT, so when we run the app now, we could able to get results when we start typing in the searchBar

Carrier
6:28 AM
Wednesday
25
Calendar
Photos
Maps
Wallet
Reminders
News
Health
Settings
Safari
Messages

The entire project can be downloaded here

• • •