**OnTheMap/AppDelegate.swift** **2**

AWESOME

Your project structure looks good, but I encourage you to have a look at 2 architecture below. Actually, I'm using Clean Swift for my current projects

* <https://hackernoon.com/introducing-clean-swift-architecture-vip-770a639ad7bf>
* <https://medium.com/@smalam119/viper-design-pattern-for-ios-application-development-7a9703902af6>

And this is the project structure I used for this project.

프로젝트 소스코드를 정리할 때, 그룹을 만들어서 정리하면 좋다!

텍스트, 지도이(가) 표시된 사진

자동 생성된 설명스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

SUGGESTION

You can remove these empty functions. They don't make any difference and your code looks clean.

**OnTheMap/Controller/SpotMapViewController.swift** **1**

스크린샷이(가) 표시된 사진

자동 생성된 설명

You can follow this example to check network before sending the request to the server

<https://stackoverflow.com/questions/30743408/check-for-internet-connection-with-swift>

**OnTheMap/Model/StudentLocation.swift** **1**

AWESOME

struct Studentlocation: codable { }

**Tips:**

There are 2 ways to create a struct and parse jsonObject to struct

* using init() method
* extend Codable

**External Resources**

* <https://medium.com/@sergueivinnitskii/easy-struct-initialization-in-swift-8ee46b8d84d5>
* <https://hackernoon.com/everything-about-codable-in-swift-4-97d0e18a2999>

**OnTheMap/Controller/MapViewController.swift** **1**

//

// MapViewController.swift

// OnTheMap

//

// Created by 강선미 on 26/07/2019.

// Copyright © 2019 Yoshimi. All rights reserved.

//

import UIKit

import MapKit

class MapViewController: UIViewController {

@IBOutlet weak var mapView: MKMapView!

@IBOutlet weak var activityIndicator: UIActivityIndicatorView!

var annotations = [MKPointAnnotation]()

override func viewDidLoad() {

super.viewDidLoad()

self.mapView.delegate = self

}

override func viewDidAppear(\_ animated: Bool) {

super.viewWillAppear(animated)

refreshMap(animated)

}

@IBAction func refreshMap(\_ sender: Any) {

pullData()

}

@IBAction func addSpotTapped(\_ sender: Any) {

activityIndicator.startAnimating()

let alertVC = UIAlertController(title: "Warning!", message: "You've already put your pin on the map.\nWould you like to overwrite it?", preferredStyle: .alert)

UdacityClient.getStudentLocation(singleStudent: false, completion:{ (data, error) in

guard let data = data else {

print(error?.localizedDescription ?? "")

return

}

if data.count > 0 {

alertVC.addAction(UIAlertAction(title: "Yes", style: .default, handler: { [unowned self] (\_) in

self.performSegue(withIdentifier: "addSpot", sender: (true, data))

}))

alertVC.addAction(UIAlertAction(title: "No", style: .default, handler: nil))

self.present(alertVC, animated: true, completion: nil)

} else {

self.performSegue(withIdentifier: "addPin", sender: (false, []))

}

})

self.activityIndicator.stopAnimating()

}

## Tips:

You can add a Navigation Bar to the TabBar and add a TabBarViewController to that TabBar.  
With the buttons in the TabBar you can process them in the TabBarViewController

class TabBarLocationsViewController: UITabBarController {

@IBAction func onClickRefreshButton(\_ sender: Any) {

onRefreshData()

}

@IBAction func onClickAddButton(\_ sender: Any) {

}

}

[스크린샷이(가) 표시된 사진

자동 생성된 설명](https://udacity-reviews-uploads.s3.us-west-2.amazonaws.com/_attachments/138757/1528426491/Screen_Shot_2018-06-08_at_09.53.42.png)