

## Tips:

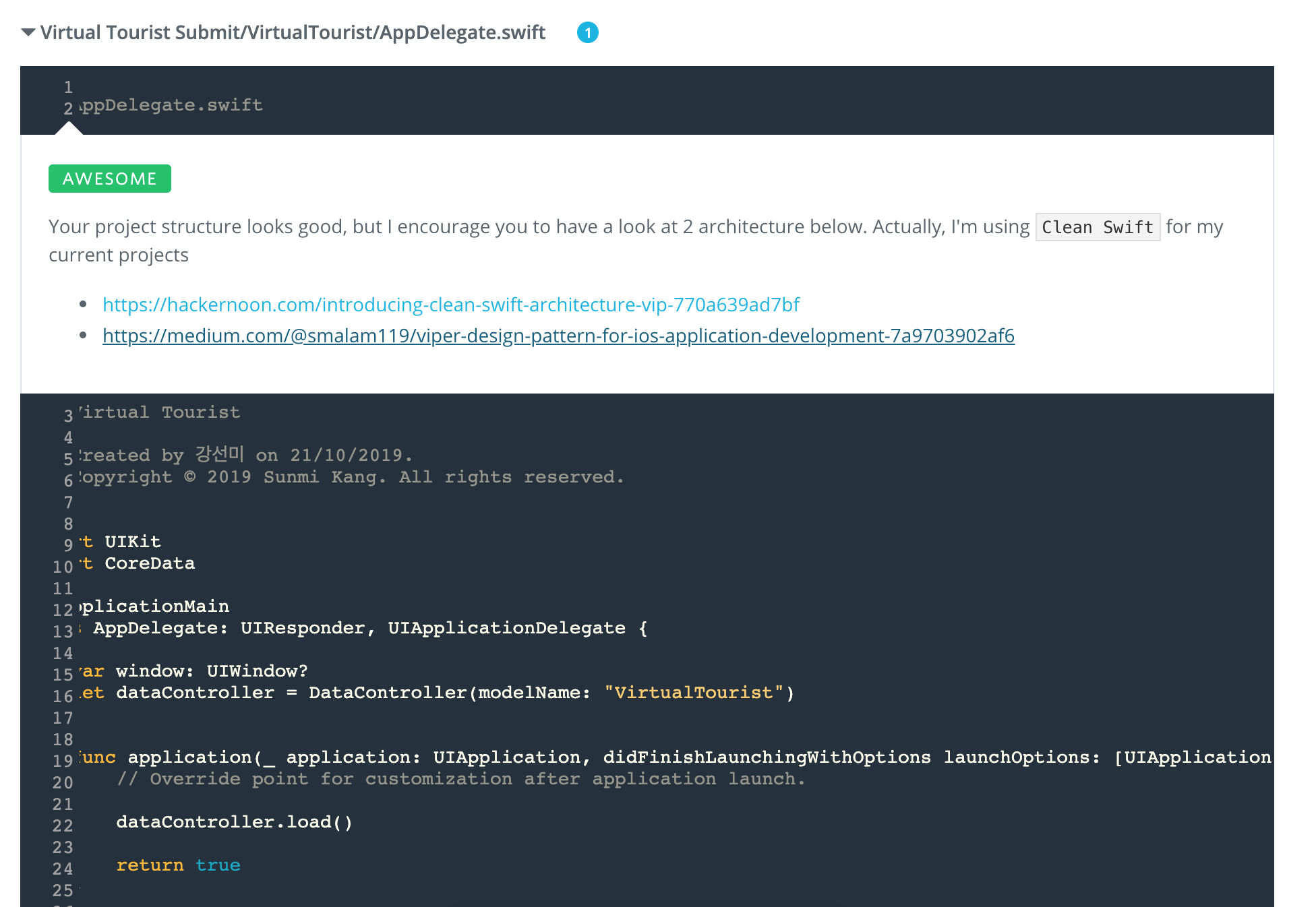
Alamofire is my favorite library for networking in iOS, you can consider and use it in the next projects

**External Resources**

<https://medium.com/@AladinWay/write-a-networking-layer-in-swift-4-using-alamofire-and-codable-part-1-api-router-349699a47569>

<https://github.com/Alamofire/Alamofire>

**Virtual Tourist Submit/VirtualTourist/AppDelegate.swift**

****

<https://hackernoon.com/introducing-clean-swift-architecture-vip-770a639ad7bf>

<https://medium.com/@smalam119/viper-design-pattern-for-ios-application-development-7a9703902af6>

**Virtual Tourist Submit/VirtualTourist/Model/DataController.swift**

SUGGESTION

You can use Singleton Pattern for DataController like this

class DataController {

let persistentContainer:NSPersistentContainer

var viewContext:NSManagedObjectContext {

return persistentContainer.viewContext

}

let backgroundContext:NSManagedObjectContext!

init(modelName:String) {

persistentContainer = NSPersistentContainer(name: modelName)

backgroundContext = persistentContainer.newBackgroundContext()

}

func configureContexts() {

viewContext.automaticallyMergesChangesFromParent = true

backgroundContext.automaticallyMergesChangesFromParent = true

backgroundContext.mergePolicy = NSMergePolicy.mergeByPropertyObjectTrump

viewContext.mergePolicy = NSMergePolicy.mergeByPropertyStoreTrump

}

func load(completion: (() -> Void)? = nil) {

persistentContainer.loadPersistentStores { storeDescription, error in

guard error == nil else {

fatalError(error!.localizedDescription)

}

// self.autoSaveViewContext()

self.configureContexts()

completion?()

}

}

static let shared = DataController(modelName: "VirtualTourist")

}

So you only need to load it in AppDelegate

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

// Override point for customization after application launch.

DataController.shared.load()

return true

}

And each time you want to access DataController you only need to call this Singleton Class

let pin: Pin = Pin(context: DataController.shared.viewContext)

pin.latitude = String(pinAnnotation.coordinate.latitude)

pin.longitude = String(pinAnnotation.coordinate.longitude)

DataController.shared.saveContext()