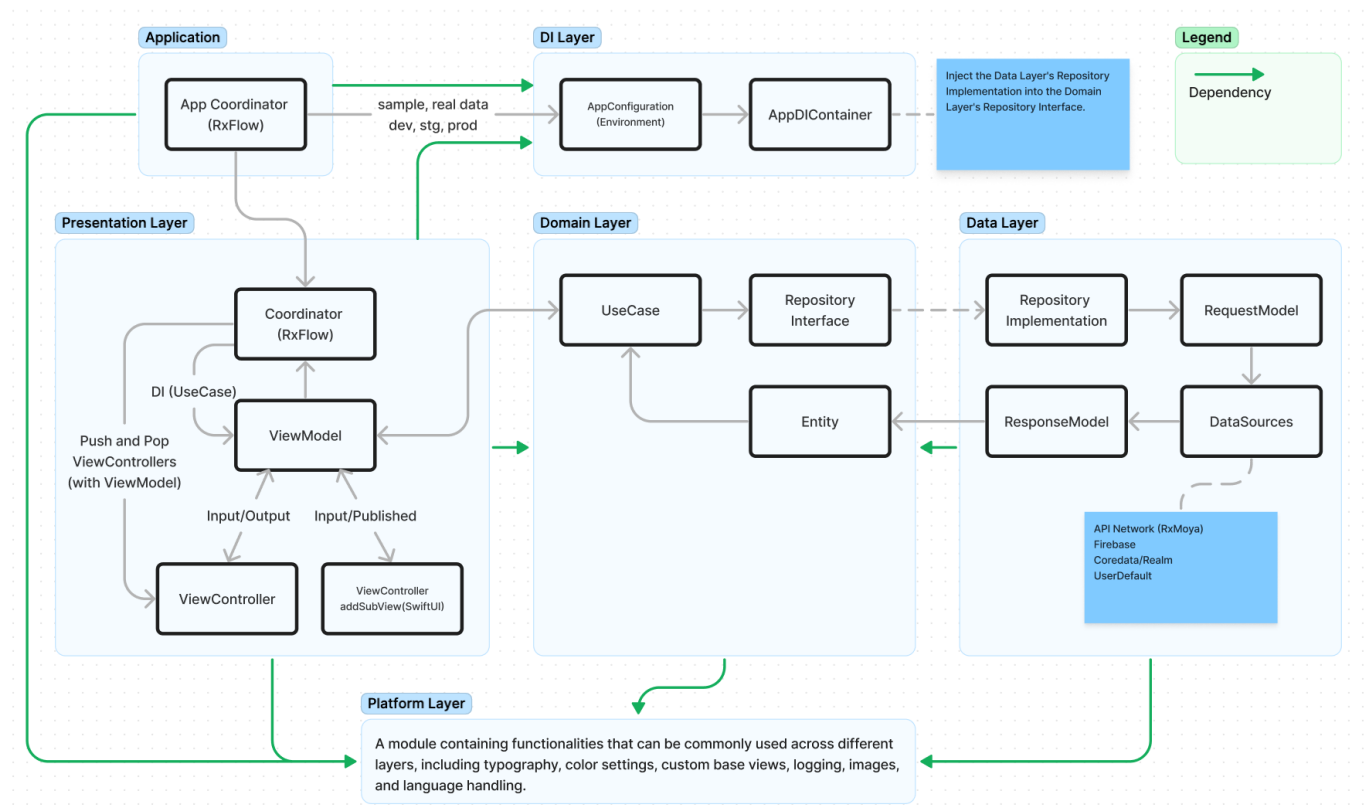


# Clean Architecture + MVVM + RxSwift + RxFlow + Moya

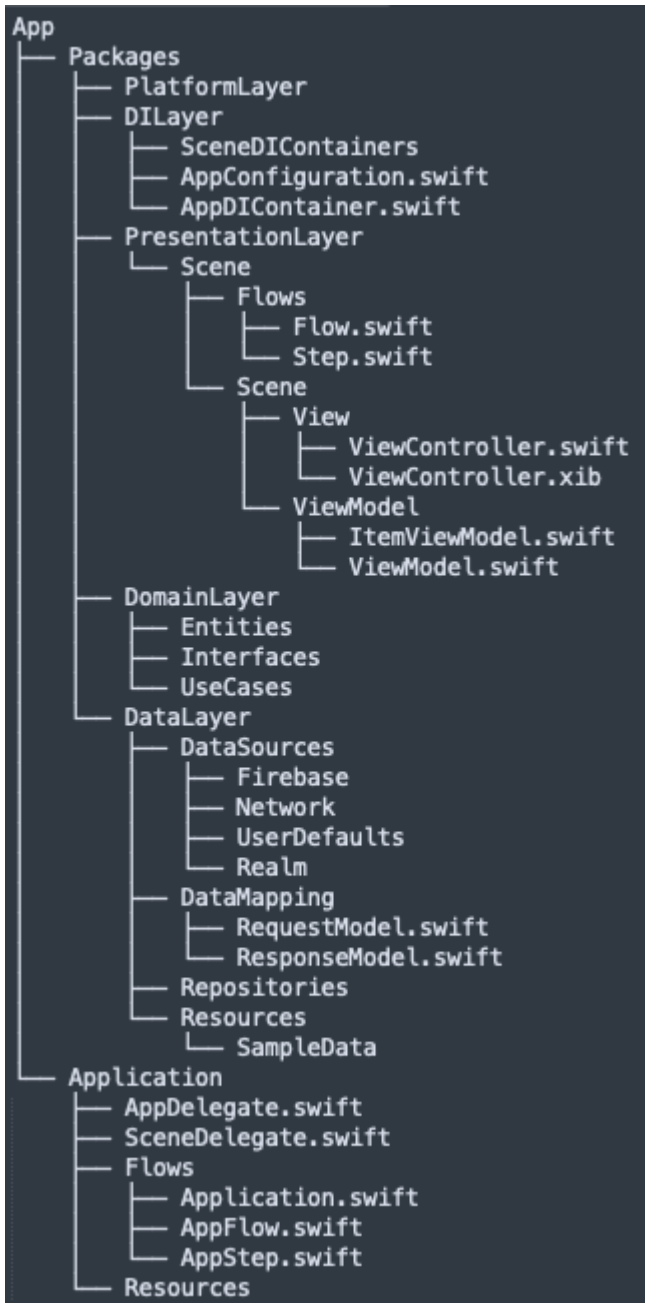
Clean Architecture + MVVM + RxSwift + RxFlow + Moya로 프로젝트를 구성하였습니다.

이 프로젝트를 참고해서 RxFlow 추가 및 Package 분리, 테스트 케이스 추가를 하였습니다. [iOS-Clean-Architecture-MVVM-RxSwift](#)

## 전체 Architecture 구성도



## 간략 File 구성도



## 구현

- xib에서 swiftui로 넘어가는 과정에서 사용하기 적합하게 구현
- MVVM 패턴에서 Input, Output을 이용해서 RxSwift로 구현
- SwiftUI에서는 MVVM 패턴에서 Input, Published를 이용해서 RxSwift로 구현
- RxSwiftExt를 이용해서 TableView의 마지막까지 스크롤하게 될 경우 다음 페이지 로드 구현
- RxMoya를 통해 API Network 구현
- RxSwift로 테이블 구현
- Code-base UI의 경우 SnapKit, Then을 사용
- Kingfisher를 이용해서 이미지 비동기 로드 및 TableViewCell이 재사용되면서 이전 이미지가 로드 될 수 있는 경우에 대해 처리
- SwiftLint를 이용해서 코드 컨벤션 유지

## Application

- Target이나 define을 활용해 환경을 DI Layer로 전달, 모의 객체(Mock)을 사용할 건지 API 주소는 어떤걸 사용할 건지 정함
- Splash 화면을 새로운 윈도우로 잠시 띄우게 처리, 기존 윈도우에서는 메인 화면을 미리 그리도록 처리, 추후 API 통신 이후 해당 화면이 사라지게 처리 가능

## DI Layer

- 객체 생성: 필요한 객체를 생성하고 초기화, 이를 통해 각 객체는 필요한 의존성을 주입받아 사용
- 의존성 관리: 각 컴포넌트와 모듈이 필요로 하는 의존성을 관리, 이를 통해 의존성이 변경되거나 업데이트되는 경우, DI Layer에서만 변경하면 됨
- 객체 재사용: 생성한 객체를 여러 곳에서 재사용 가능, 이는 메모리 사용량을 줄이고 성능을 향상시킴
- 테스트 용이성: 테스트에서 실제 구현체 대신 모의 객체(Mock)를 주입하여 테스트를 쉽게 수행, 이를 통해 테스트가 더욱 견고하고 안정적이게 됨

## Platform Layer

- 각 레이어에서 공통으로 사용할 수 있는 기능들을 모아 놓은 계층
- 타이포그래피, 색상 설정, 커스텀 기본 뷰, 로그, 이미지, 언어 처리 등이 포함

## Presentation Layer

- 사용자 인터페이스와 관련된 요소를 처리하는 계층
- 사용자의 입력을 받아들이고, 애플리케이션의 상태를 화면에 표시하는 데 사용

## Domain Layer

- 비즈니스 로직과 애플리케이션의 핵심 기능을 처리하는 계층
- 이 계층은 도메인 객체, 비즈니스 규칙 및 사용 사례를 정의
- Platform Layer를 제외한 다른 레이어와의 의존성이 없도록 구현

## Data Layer

- 데이터 소스와의 상호 작용을 처리하는 계층
- 데이터를 저장하고 검색하는 로직이 포함되어 있으며, 원격 API, 데이터베이스, 파일 시스템 등과 같은 다양한 데이터 소스와 통신
- 이 계층은 Domain Layer가 요청하는 데이터를 제공하며, 데이터 저장소의 구현 세부 정보를 추상화

## Coordinator with RxFlow

- 각 Scene에 맞춰서 RxFlow를 분리
- SceneDIDContainer와 UINavigationController를 초기화 메서드에서 받아 처리
- SceneDIDContainer로 ViewModel에 UseCase를 만들어서 주입
- UINavigationController로 화면 이동은 push로만 처리
- 해당 Scene이 끝나는 경우 RxFlow의 .end(forwardToParentFlowWithStep:)가 호출되게 하여 해당 Scene의 Flow가 종료되게 처리

## Built With

- [RxSwift](#) - Reactive Programming in Swift

- [RxCocoa](#) - library allows us to use Cocoa APIs used in iOS and OS X with reactive technics.
- [RxSwiftExt](#) - A collection of Rx operators & tools not found in the core RxSwift distribution
- [Moya](#) - Network abstraction layer written in Swift.
- [Kingfisher](#) - A lightweight, pure-Swift library for downloading and caching images from the web.
- [SnapKit](#) - SnapKit is a DSL to make Auto Layout easy on both iOS and OS X.
- [SwiftLintPlugin](#) - A Swift Package Plugin for SwiftLint that will run SwiftLint on build time and show errors & warnings in Xcode.
- [Then](#) - ✨ Super sweet syntactic sugar for Swift initializers

## TO-DO

- Deep Link가 가능하도록 구현 필요
- Firebase, Realm, Userdefaults 처리 구현 필요