

Movie app

Movie app shows the latest released movies in the world. The endpoint contains the filter by release date.

Architecture

I used MVVM for the architecture. It Separates presentation logic and business logic from the view improving the testability of the code.

For the binding between the view and view model I used closures for simplicity. The best approach would be use maybe FRP like RxSwift.

Unit Tests

I use the *dependency injection* technique to design my **MovieListViewModel**. The property *apiService* is a dependency of the **MovieListViewModel**. The *apiService* could be assigned by all objects conforming the **APIServiceProtocol**. In the test enviroment we inject a *mockApiService*, an object only designed for the test.

I tested the **MovieListViewModel** and the **APIService**, because they are the most important parts of the app. But also the *ImageLoader* and the search controller methods could have been tested. Due to time problems I did not test them.