

This app is a client for the The Movie DB (TMDB) API. It allows you to view different types of movie and tv info as well as “favorite” certain content. More info on the TMDB API can be found at <https://www.themoviedb.org/documentation/api>.

The app was developed using RxSwift, Core Data and the MVVM design pattern.

Architecture

The project can be broken down into the following folders:

1. Controllers – For all view controllers.
2. Network – For any networking related classes. (ex The Movie DB API client, HTTP requests)
3. Resources/Assets – Image assets.
4. Storyboard – The main app storyboard.
5. Models – The data models and persistence layer.
6. ViewModel – The logic and domain layer.
7. Views – Any custom views

The app has 3 main view controllers. Each view controller subscribes to RxSwift observables in its view model to get its data.

- Movies View Controller: Displays the collection of movie and tv content and allows searching.
- Movies Detail View Controller: Displays information about a specific movie.
- Tv Detail View Controller: Displays information about a specific tv show.

There are also 3 corresponding view models that are responsible for using the network layer to query data and allow it to be bound to the view observers. Each view model is owned by a view controller and doesn't know of the view controller's existence.

- Movies View Model: Uses the network layer to get movie and tv data and can use the persistence layer to store and retrieve data.
- Movie Detail View Model: Uses the network layer to get movie detail data and makes it available to possible observers.
- TV Detail View Model: Uses the network layer to get tv show detail data and makes it available to possible observers.

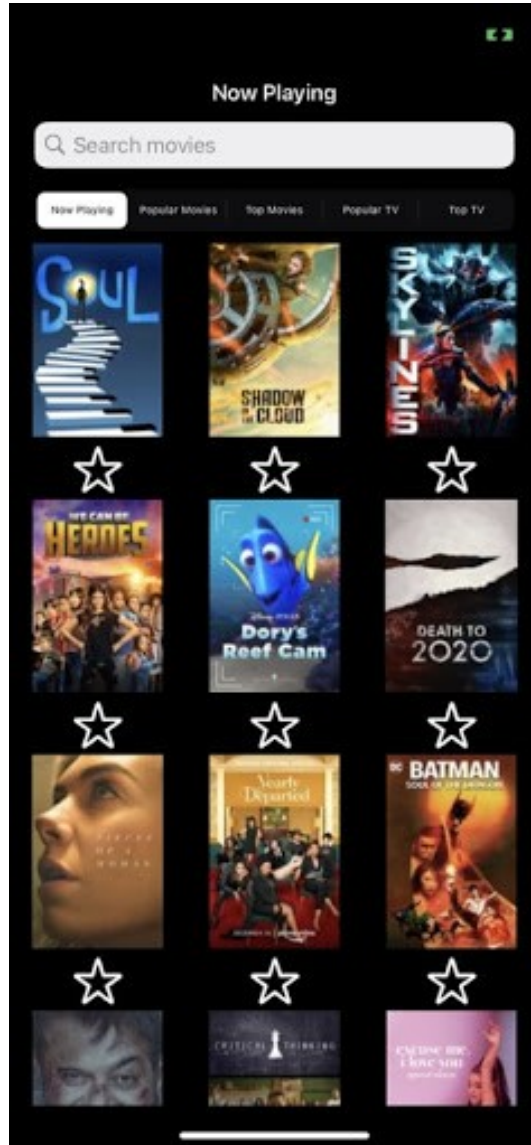
The network layer consists of an API client responsible for constructing queries for The Movie DB API, a network manager that is responsible for sending HTTP requests and decoding JSON, and a network connectivity manager that monitors for changes in the network connection.

The persistence layer consists of structs to represent the data returned from the network layer, a data manager responsible for persisting data using Core Data, and an image cache manager responsible for reading and writing images to disk.

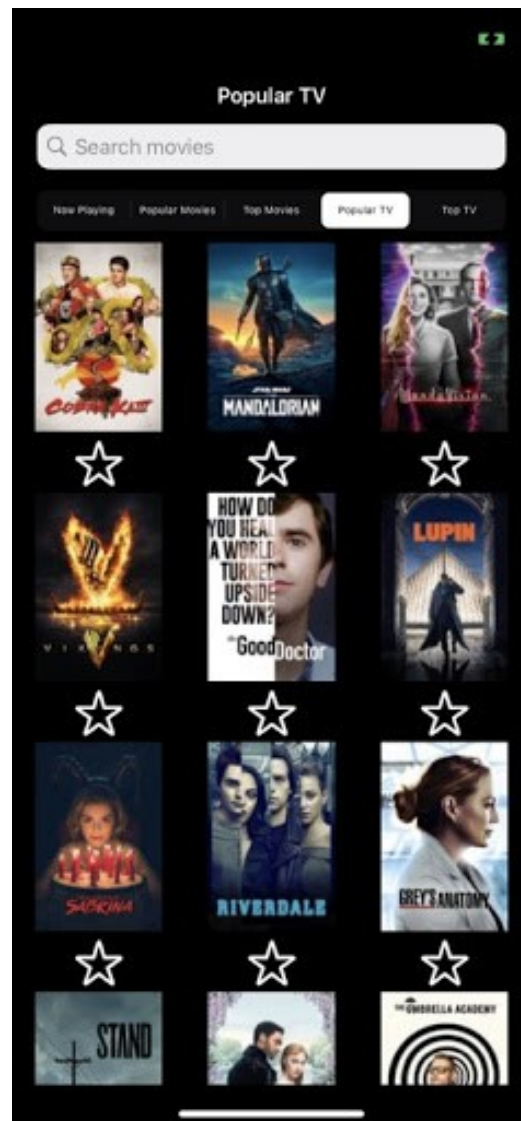
View controllers, view models, and network and persistence dependencies never interact directly, only through their protocols. This ensures loose coupling so long as different implementations implement the given protocol. This particularly was useful for making “mocks” of the persistence layer and network layer for unit testing. A view model can be instantiated with a “mock” data manager and API that can respond just like its real counterpart, without having to write to disk or make a network connection.

User Guide

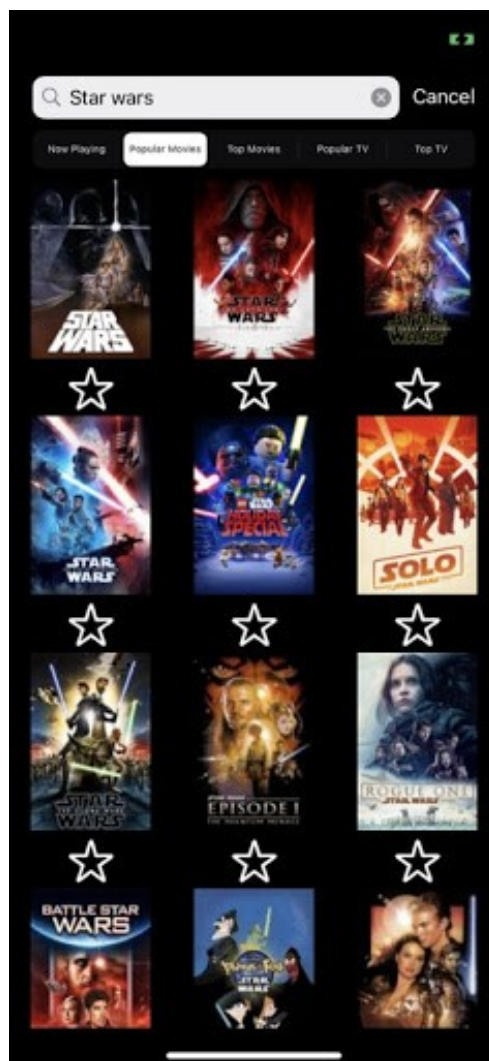
On app start, the Movies View Controller will display with the last featured category of movies or tv content being displayed. If this is the first time the app was launched and the cache is empty, then movies that are “Now Playing” are displayed.



From here you can tap on different featured categories in the scope bar to discover other movie or tv content.



The search bar allows you to query movie content only. Tapping cancel will close the search and take you back to the last selected feature category.



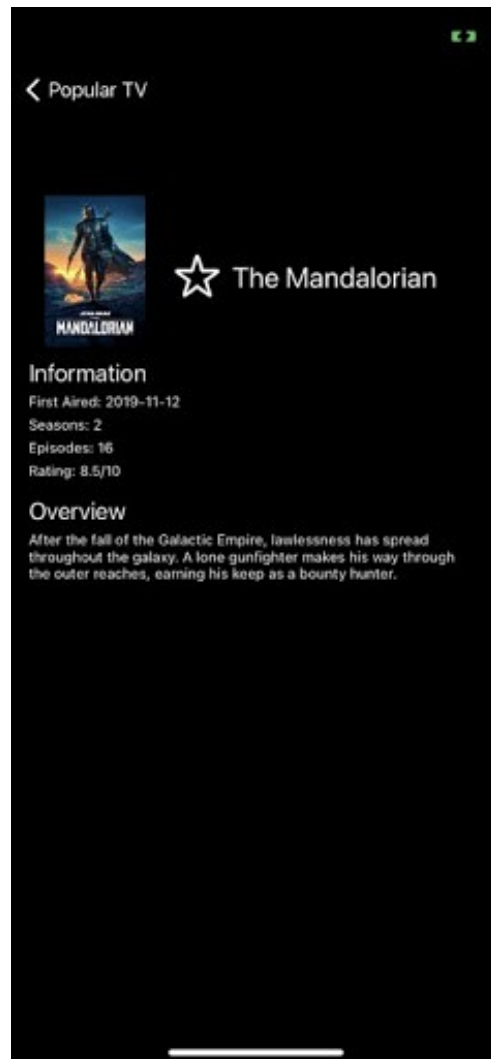
Tapping a “star” under a movie or tv show, will favorite that content. The next time you come across that content, the filled in star will indicate that you had favorited it.



Tapping on a movie displays more in depth movie details. You can view info such as the title, running time, voter average rating out of 10, the release data and an overview of the movie. You can also “favorite” content in this view as well.



Returning to the previous view, you can switch to a TV related featured category and tap on a TV show to display more in depth TV show details. You can view info such as the title, the date the show first aired, how many seasons and episodes there are, the voter average rating out of 10, and an overview of the show. You can also “favorite” content in this view as well.



In case of loss of network, if you open the app from being closed, then the last selected feature is displayed from the cache, with the search and navigation bar disabled. Detail content is also disabled. Restoring the network connection re-enables all of the controls.

