

Project Files and Their Responsibilities

- **MoviesListViewController**
It is main page. It consists of a search bar and collection view.
- **MovieCollectionViewCell**
It is a cell view that used in collection view.
- **MovieDetailViewController**
When select an item in collection view, this viewcontroller is pushed. It is detail page of selected movie.
- **MoviesResponse**
This struct correspondings to return value of movie API.
- **Movies**
This struct correspondings to each element inside results array in MoviesResponse
- **MovieDetail**
This struct correspondings to return value of movie detail API.
- **MovieServiceAPI**
This class is a Singleton class that is used for calling services. It consists of two methods that call movie API and movie detail API. It is possible to write a generic method instead of two different methods, but I haven't written because there are just two services.

Project Overview

I haven't used storyboards in the project. Instead, I've used interface builder(xib) files. In SceneDelegate.swift file, I set the rootViewController to a UINavigationController. It's rootViewController is MovieListViewController. So, when app starting, MovieListViewController becomes the top.

In MovieListViewController, there is a search bar and collection view. By calling *fetchAPIDataAndSetToCollectionView* method, movie rest api called, and set returned data to *filteredMoviesList* and *moviesList* array. I created two arrays because one of them is to show filtered data during search bar operation, other one is to protect real data. As same approach, there are *filteredFavData* and *favData* arrays to keep data that a movie is in favorite or not.

When selected a cell in collection view, *MovieDetailViewController* pushed to navigation controller. Here, there is a favorite button to add related movie as favorite. I have used UserDefaults option to saved data on device, as movie id is key, and boolean data is value. When user clicks the favorite button, UserDefaults operations happened in *addOrRemoveFavorite* function.

I have handled paging operation in *willDisplay* method of collection view. I have used *isWaiting* flag in order to prevent call service consecutively.