

Tubi TV iOS Engineer Coding Challenge

Project Overview:

Given an endpoint that returns a list of movies in JSON format, create a native iOS app that displays returned data in a collection view or table view. When the user taps on one of the cells, the app should open a detail view for the corresponding movie and fetch the movie details if needed. Once movie details fetching is done, it should display movie details on the details view. Movie details should be cached in a LRU cache (least recently used cache).

Time Limit:

There is no limit =)

Feature Requirements:

- Fetch movies list data, display it in a collection view or table view
- When user taps on a cell, the app should open the movie details view
 - Please use the movie id from the movies list to fetch the movie details
 - Please display index field from movie details in your detail view
- Please implement your own LRU (least recently used) cache for movie details
 - Please don't use any open source 3rd party library or NSCache
 - No need to worry about thread-safe (only consider UI thread operations)
 - Cache should be able to store any types of data (use generic types)
 - Maximum size should be 5
 - The cache should support add, get, isValid. All operations must be $O(1)$
 - add: add one item to cache
 - get: get one item based on key from cache
 - isValid: check if a item is still valid based on the key
- Once movie details view is displayed, the app should check LRU cache first
 - If cache exists, use cache to display movie details
 - If cache doesn't exist, fetch data from movie details endpoint. After getting data from endpoint, display movie details
- Unit tests

API Endpoint:

- Movies list:
 - <https://us-central1-modern-venture-600.cloudfunctions.net/api/movies>
- Movie details:

- https://us-central1-modern-venture-600.cloudfunctions.net/api/movies/{movie_id}

Programming Language/SDK Usage:

- It's recommended to use Swift, but you can also use Objective C
- Native iOS UIKit
- LRU Cache has to be your own implementation, so no 3rd party library or NSCache for caching
- It's OK to use 3rd party networking and image frameworks like Alamofire, SDWebImage

Grading:

- Code quality, cleanness, efficiency and reusability
- Error handling, and edge case scenario fallback design
- Overall code architecture design and extendibility

Project Submission:

- Please submit the source code in a zip file and maintain Xcode project structure.
- The project should be able to unzip and be able to run without any structural integrity issues.