iOS Code Challenge: Movie Search

You are asked to implement an iOS app that will search for movies through The Movie Database (TMDb). The app will allow us to type in a keyword and then visualize all the movies matching that keyword. For each movie returned, the app will display its title, poster, and overview (short summary of the plot).

Use the “Search” API provided by https://www.themoviedb.org/documentation/api

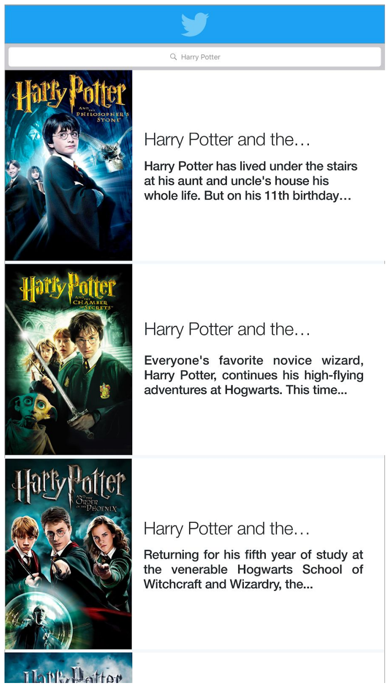
- Make a GET request to https://api.themoviedb.org/3/search/movie with the following parameters:
  - `api_key`: for convenience, use `2a61185ef6a27f400fd92820ad9e8537`
  - `query`: a string representing the keyword we are searching for
- Make a GET request to https://image.tmdb.org/t/p/w600_and_h900_bestv2/{poster} to download the poster image for the movie, where `{poster}` is the value returned by the previous endpoint within the `poster_path` field.

Sample Requests

- Searching for “Harry Potter”: https://api.themoviedb.org/3/search/movie?api_key=2a61185ef6a27f400fd92820ad9e8537&query=Harry%20Potter
- Harry Potter and the Philosopher’s Stone poster: https://image.tmdb.org/t/p/w600_and_h900_bestv2/lR4drT4VGfts32j9jYTZUc1a3Pa.jpg

Hard Requirements

- The code needs to compile and run in the simulator
- The project should run out of the box in the latest (non-beta) version of Xcode
  - No modifications required by the evaluator
  - No third party tools or libraries
- The code should not crash under normal usage (searching, scrolling, rotation)
- The code should contain no third party libraries
  - NSCache, NSJSONSerialization, NSURLSession, and UISearchController are your friends
- The code should be performant
  - Thread management
  - In-memory image caching
  - Table cell reuse
  - Placeholder for unloaded images


Harry Potter

Harry Potter and the…

Harry Potter has lived under the stairs at his aunt and uncle's house his whole life. But on his 11th birthday…

Harry Potter and the…

Everyone's favorite novice wizard, Harry Potter, continues his high-flying adventures at Hogwarts. This time...

Harry Potter and the…

Returning for his fifth year of study at the venerable Hogwarts School of Witchcraft and Wizardry, the...

Other Considerations

- The code should be easy to review
    - Be consistent in coding style
    - Write comments where necessary
- Write unit tests to cover core business logic

Not Required

- This project is meant to be narrow in scope, you should not implement:
    - Pagination
    - Disk-based cache
    - UI automation tests
- Your project only needs to support the latest (non-beta) build target

Notes

- You are not required to follow the exact design sample provided by us
- Objective-C and Swift code are welcome
- You can use Apple's Network Link Conditioner to test your app under slow network conditions to test proper caching and cell reuse
- While time to complete a solid answer varies, you should expect to spend around 3-5 hours on the project
- When submitting, email us back a compressed zip file of your project and briefly outline what you would implement, if you were given more time, with the intention of submitting to the App Store
- Please do not share this document publicly
- Please reach out to us if you have questions