





 [LambdaSchool](#) / [ios-sprint4-challenge](#)

Sprint challenge for Sprint 4 of Lambda's iOS course

 6 commits

 1 branch

 0 releases

 2 contributors

Branch: master ▾


New pull request

Create new file




Upload files


Find file

Clone or download ▾

 **SpencerCurtis** Update README.md

Latest commit 2324777 on Aug 17, 2018

 <a href="#">MyMovies</a>	Update project to match instructions	5 months ago
 <a href="#">.gitignore</a>	Initial commit	7 months ago
 <a href="#">README.md</a>	Update README.md	5 months ago

 **README.md**

# MyMovies

---

## Instructions

Please read this entire README to make sure you understand what is expected of you before you begin.

This sprint challenge is designed to ensure that you are competent with the concepts taught throughout Sprint 4.

In your solution, it is especially important that you follow best practices such as MVC and good, consistent code style. You will be scored on these aspects as well as the project requirements below.

Begin by forking this repository. Clone your forked repository to your machine. Use the provided Xcode project in the repository as it contains starter code. Commit as appropriate while you work. Push your final project to GitHub, then create a pull request back to this original repository.

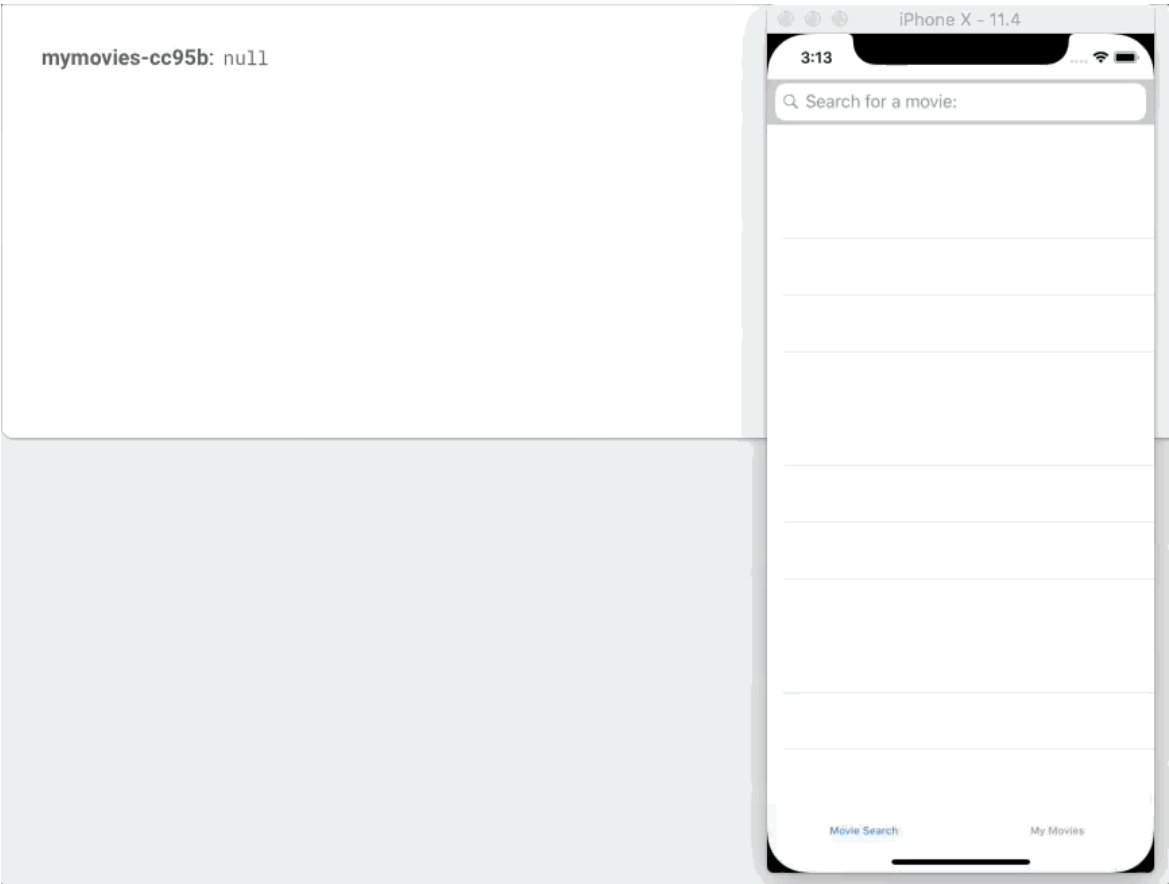
**You will have 3 hours to complete this sprint challenge**

Good luck!

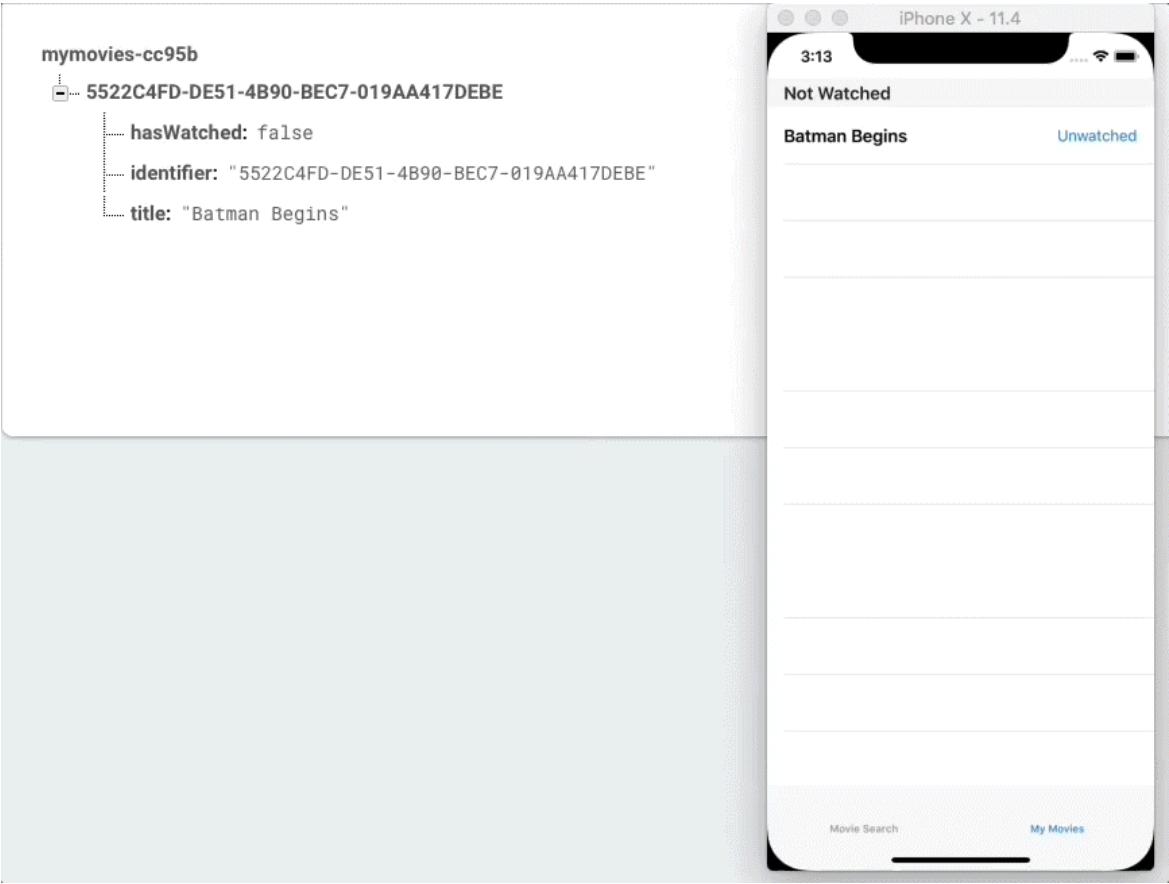
## Screen Recordings

Please view the screen recordings so you will know what your finished project should look like:

Adding, updating, and deleting movies to/from Core Data and Firebase:



Syncing between Firebase and Core Data:



(The gifs are fairly large in size. It may take a few seconds for them to appear)

## Requirements

---

This project uses The Movie DB API. The starter project has search functionality working for you already, as that isn't the point of this Sprint Challenge. There is no need to reference the API's documentation whatsoever.

The requirements for this project are as follows:

1. A `Movie` Core Data object. Its attributes should match the properties found in the `MovieRepresentation` object in the starter project.
2. Update the `MovieSearchTableViewController` 's prototype cell to have a button that allows the user to save a movie to their list of saved movies. These saved movies should be stored in Core Data.
3. Display the list of saved movies in the `MyMoviesTableViewController` . You must use an `NSFetchedResultsController` to display the movies. Separate the movies into two sections in the table view by whether they have been watched or not.
4. This table view's prototype cell should let the user update whether they have seen the movie or not.
5. Send changes to the user's saved movies to a Firebase Database when creating, saving, and deleting movies.
6. Synchronize the movies in Firebase with the device's `NSPersistentStore` . This must be done on a background `NSManagedObjectContext` .