IvanCaldwell / **ios-sprint7-challenge**
forked from LambdaSchool/ios-sprint7-challenge

No description, website, or topics provided.

Edit

Manage topics

5 commits	1 branch	0 releases	1 contributor

Branch: master ▾   New pull request

Create new file   Upload files   Find file   Clone or download ▾

This branch is even with LambdaSchool:master.   Pull request   Compare

armadsen Add GIF to README   Latest commit 53164a1 on Sep 6, 2018

Random Users.xcodeproj	Add starter project	5 months ago
Random Users	Add starter project	5 months ago
.gitignore	Initial commit	7 months ago
README.md	Add GIF to README	5 months ago

README.md

Random Users

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 7, Intermediate Swift.

Begin by forking this repository. Clone your forked repository to your machine. Use the provided Xcode project in the repository. It includes `ConcurrentOperation` for you to use, as well as a placeholder Lambda logo image you can use if you'd like. 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

If you have any questions about the project requirements or expectations, ask your PM or instructor. Good luck!

Screen Recording

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

(The gif is fairly large in size. It may take a few seconds for it to appear)

Requirements

The goal of this challenge is to build a simple "contact manager" app driven by the randomuser.me API. The randomuser.me API provides lists of random, fake people on demand. See the documentation for full information. In short, do a GET request to https://randomuser.me/api/?format=json&inc=name,email,phone,picture&results=1000 to get JSON with results back. You can change the `results` URL query parameter to control the number of users you get back (max is 5000).

The requirements for this project are as follows:

1. Your main view controller should display a table view with loaded users.
2. Each table view cell must display the user's (thumbnail) image and their name.
3. The app must fetch and display at least 1000 users. You may include more if desired. You can also add a button to continue loading more users if you'd like (not required).
4. Tapping a user should display a detail view with their full-size image and details including name, phone number, and email address.
5. Your app must fetch images on-demand as the corresponding row is scrolled on screen.
6. It must cancel no-longer-needed fetches as rows scroll off screen.
7. Scrolling must be smooth and responsive regardless of the number of loaded users.
8. You must implement a generic `Cache` to cache already-fetched thumbnail and fullsize images. `Cache` **must** be thread-safe.