LambdaSchool / **ios-sprint3-challenge**

Sprint

7 commits | **1** branch | **0** releases | **3** contributors

Branch: **master** ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

erica Got rid of image scaling. All pokemon sprites are tiny | Latest commit `acc7a2a` 15 hours ago

.gitignore	Initial commit	5 months ago
README.md	Got rid of image scaling. All pokemon sprites are tiny	15 hours ago

README.md

Networking Sprint Challenge: Pokedex

This Sprint explored material related to networking and asynchronous tasks on iOS. During this Sprint, you studied dispatch, REST, URL Sessions, completion handlers and more. In your challenge this week, you demonstrate proficiency in these topics by creating an application that uses a free Pokemon API to allow users to search for their favorite monsters.

Instructions

Read these instructions carefully. Understand exactly what is expected *before* starting this Sprint Challenge.

Follow best practices to ensure your solution is consistent and well designed. You are scored on these aspects as well as meeting the project MVP (minimum viable product) requirements listed below.

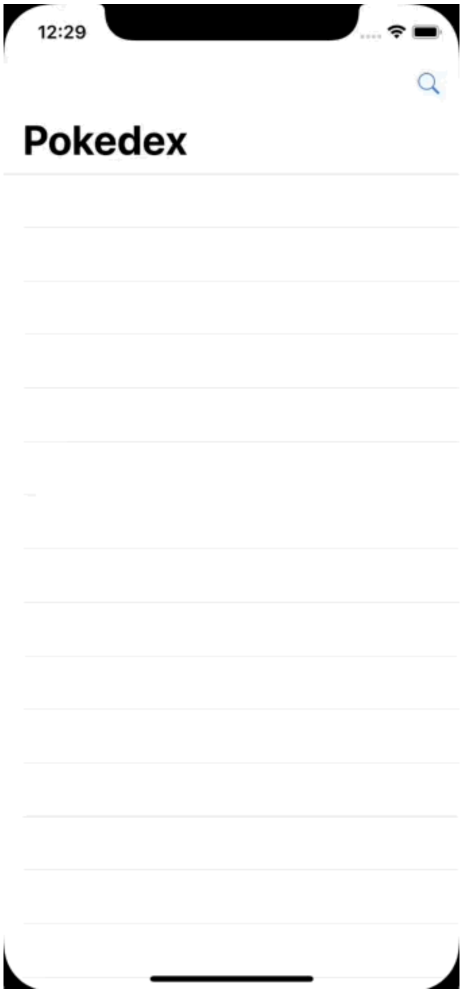
Fork this repository and clone your fork to your computer. Create your Xcode project in this cloned fork repository folder. Commit appropriately as you work. When finished, push your final project to GitHub and create a pull request back to this original repo.

You have three hours to complete this Sprint Challenge

Good luck!

Screen Recording

This screen recording previews the look and feel of a completed project that meets the MVP requirements for this project. Please be patient. This GIF is large and may take several seconds to load:



Requirements

This project uses the **pokeapi.co API** (API Documentation). Use this API to search for pokemons and retrieve their information.

The requirements for this project are as follows:

1. Establish a `Pokemon` record with properties for the Pokemon's name, ID, ability, and types.
2. Display a list of matching Pokemons and show the sprite (image) associated with the record.
3. Allow the user to save the Pokemon. Display a list of saved Pokemons in a table view, including the Pokemon sprites. (**Stretch:** Implement persistence for this list.)
4. Allow the user to view a Pokemon's information in a detail view controller.
5. Allow users to remove saved items from their Pokemon list. (If you implement persistence as a stretch goal, keep the saved list in sync.)
6. *Stretch*: Allow the user to search for a Pokemon using partial name matching. You will have to download a full list of the Pokemon names from the API and then perform your searches locally.
7. *Stretch*: Sort the Pokemons by their name or ID in the saved pokemon list. Offer the user a toggle that enables them to switch between the two sorts.