

iOS Proficiency Exercise

Overview

The purpose of this exercise is to assess candidate developer's iOS coding knowledge and style. The exercise involves build a “proof of concept” app which consumes a REST service and displays photos with headings and descriptions. The exercise will be evaluated on coding style, understanding of programming concepts, choice of techniques, and also by the developer's process, as indicated by the trail of git commits.

Specification

Create a universal iOS app which:

1. Ingests a json feed from

<https://dl.dropboxusercontent.com/s/2iodh4vg0eortkl/facts.json>

2. Please use Swift 4 encoding for JSON decoding/encoding
3. The feed contains a title and a list of rows
4. Displays the content (including image, title and description) in a table
5. The title in the navbar should be updated from the json
6. Each row should be the right height to display its own content and no taller. No content should be clipped. This means some rows will be larger than others.
7. Loads the images lazily

8. Don't download them all at once, but only as needed
9. Refresh function
10. Either place a refresh button or use pull down to refresh.
11. Should not block UI when loading the data from the json feed.

Guidelines

1. Use Git to manage the source code. A clear Git history showing your process is required.
2. The app should perform well on a lower specification device such as an iPhone 5
3. Scrolling the table view should be smooth, even as images are downloading and getting added to the cells
4. Support both iPhone and iPad (in both orientations)
5. If threading is used, do not spawn threads manually by using `performSelectorOnMainThread`. Use GCD queues instead.
6. Comment your code when necessary.

7. Try to polish your code and the apps functionality as much as possible.
8. Commit your changes to git in small chunks with meaningful comments
9. Don't use any singletons in your submission

Additional Requirements

1. Use programmatic auto layout to layout the cells in the app
2. Support all iOS versions from the latest back to iOS9
3. Use the NSURLSession framework for your service calls