

# 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 using Swift which:

- Ingests a json feed <https://dl.dropboxusercontent.com/s/2iodh4vg0eortkl/facts.json>
- You can use a third party Json parser to parse this if desired.
- Displays the content (including image, title and description) in a table
- The title in the navbar should be updated from the json
- 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.
- Don't download all images at once, only as needed
- Refresh function, either a refresh button or use pull down to refresh.
- Should not block UI when loading the data from the json feed.
- Support all iOS versions from the latest back at least 2 versions

## Guidelines

- Use Git to manage the source code. A clear Git history showing your process is required.
- Structure your code according to industry best practice design patterns
- Do not use any .xib files or Story Boards
- Scrolling the table view should be smooth, even as images are downloading and getting added to the cells
- Support both iPhone and iPad (in both orientations) all devices including iPhoneX
- If threading is used, do no spawn threads manually use GCD queues instead.
- Comment your code when necessary.
- Try to polish your code and the apps functionality as much as possible.
- Commit your changes to git in small chunks with meaningful comments
- Feel free to use open source components via Cocoapods or Carthage if it makes sense

## Additional Requirements

- Style your code according to this style guide <https://github.com/raywenderlich/swift-style-guide>
- Use programmatic auto layout using Layout Anchors to layout the cells in the app
- Use the URLSession framework for your service calls
- Please use a TableView as the container