ReadMe

1. This app has been developed with Telstra Specifications as stated below.

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

2. This app meets all the above specified requirements.
3. Third Party Libraries used during development are below
    - ReachabilitySwift (4.3.1) - Used for network change monitoring.
    - SDWebImage (5.0.1) – Used for asynchronous image download and cache mechanism.
    - Alamofire (4.8.2) – Used for network call (REST API).
    - SwiftLint (0.31.0)– Used for Swift coding standard.
    - NAActivityIndicator (4.7.0) – Used for custom activity Indicator.
4. Additional Feature Implemented in this app:
    - Device space calculation before image download.
    - Customized activity indicator.
    - Network status monitoring.
    - Localizable Strings.
    - Check nil data in parsed JSON data.
5. Language used for app development – Swift 4.2
6. SDK used for app development – Xcode 10.1
7. App tested on iPad Air, iPhone 6S device and iPhoneX simulator with both landscape and portrait orientation.
8. Test cases has been implemented for below scenarios
    - Data should not be nil when REST API URL is correct.
    - Data should be nil when REST API URL is incorrect.
    - SDWebImage download should not return error when REST API URL is correct.
    - SDWebImage download should return error when REST API URL is incorrect.
    - JSON Parser should return parsed data without error.
    - Check nil data in parsed JSON data.
    - Correct status and message should be passed to ViewController when JSON data is parsed.
    - Correct status and message should be passed to ViewController when refresh control is called.
9. Build Instructions:
    - Open the project in Xcode 10.1
    - If the project doesn’t install then
        - Pod deintegrate
        - Pod install
        - Change the swift language for the Pods Reachability and Alamofire to swift 4.2