iOS Code Challenge

Marvel Characters App

Here at Lottoland, we love arguing who is the best comic super-hero. Too bad we don't have a cool app to explore the Marvel Universe on our iPhone.

We hope that with this code challenge you help us to find the best super-hero!

Functional Requirements

- Create an iOS application that communicates with the Public Marvel API: https://developer.marvel.com
- Network errors or API errors must be controlled, informing a user and offering a way to try again
- Persistent data: when the app gets restarted, the data that was being shown in the first screen (Characters screen) should be shown without requiring an active network connection. Only take care of data, no need to persist images locally

Characters screen

- The app should consist on a list of characters. There should be a searching mechanism in order to filter the characters by name that starts with specified letters. For example "Sp" should return "Spider-Man" among others.
- The data shown for each character is up to you, but the app should present at least this info for each character: name, description, picture
- When the user taps on a character, the app should present a list with all the character's comics.

Comics screen

- This list should at least present the following info for each comic: title, description, image.
- **OPTIONAL:** The user should be able to sort the list by title or by publication date
- **OPTIONAL:** From this list, selecting one of the comics should present extended info of the selected comic. The amount of info and how to present it is up to you.

Technical Requirements

- Use Swift 5.x to develop the entire app. Minimum iOS version: 12.0
- Follow the architecture you think is appropriate for this application.
- Avoid using third party libraries. Use native methods to implement persistence, networking and JSON parsing
- Do not use *User Defaults* exclusively for persistence solution (but can be used in combination with other methods)
- At least one complete example of Unit Testing and UI Testing
- Please use git so we can see a little of your working methods. You can share this repository with us using GitHub, BitBucket, or simply a .zip file with the .git folder included.
- Include a README explaining the architecture you followed and some of the decisions made.

Go ahead, show off your skills, we want to see what you're capable of.
Good luck and have fun!