Xamarin Proficiency Exercise

Overview

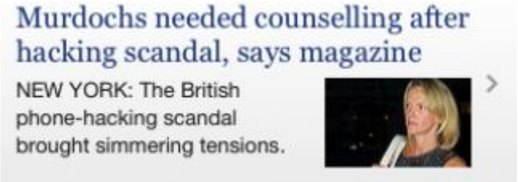
The purpose of this exercise is to assess the candidate developer's technical proficiency, coding knowledge and style. The exercise involves build a "proof of concept" app which consumes a REST service and displays photos with headings and descriptions. The exercise will be evaluated on coding style,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

Time to complete: 8 hours maximum

Specification

Create a Xamarin.Forms app which supports both iOS and Android devices:

1. Ingests a json feed from https://dl.dropboxusercontent.com/s/2iodh4vg0eortkl/facts.json
    - The feed contains a title and a list of rows
    - The title should be visible at all times at the top of the screen
2. Display the list of rows in a single scrolling view
    - Bind the data to an observable collection in a ViewModel
    - Each item contains 3 properties which need to be displayed: image, title and description
    - Each row should be dynamically sized to the correct height to display its content - no clipping, no extraneous white space etc. (this means some rows will be larger than others)
    - Each row should look roughly like the following image:



Murdochs needed counselling after hacking scandal, says magazine

NEW YORK: The British phone-hacking scandal brought simmering tensions.

3. Load the images lazily (don't download them all at once, but only as needed)
4. Implement a refresh button allowing the data & view to be updated
    - The refresh function will reload all items back into the observable collection
5. Implement a sort button that will sort the collection in alphabetical order by title
    - If pressed again, the sort order will be reversed
6. All buttons will use commands implemented in the ViewModel
    - All buttons should be visible at all times at the bottom of the screen
7. Do not block UI when loading the data from the json feed

Optional Extra Credit

1. Load the list from a file on the file system

Additional Guidelines

1. Use a Github repository to manage the source code. A clear Git history showing your process is required. Commit your changes to git in small chunks with meaningful comments.
2. The list should scroll smoothly. As much data as possible should be cached.
3. Comment your code where necessary.
4. Polish your code as much as possible - we expect professional, production quality code.
5. Feel free to use any best-practice open-source libraries/examples you need, just be sure to give credit. Do not use beta versions of any libraries.
6. Include at least two UI unit tests; one that asserts the state of the screen when set up with all data present, and one that asserts the state of the screen when in an error state.
7. Handle screen rotation efficiently.