

Coding Challenge

Requirements

- Write a mobile app that uses the Flickr image search API and shows the results in a 3-column scrollable view.
- The app must let users enter queries, such as "kittens".
- The app must support endless scrolling, automatically requesting and displaying more images when the user scrolls to the bottom of the view.
- Do not use third-party libraries. They should not be needed for a project of this scale and we want to make sure you are familiar with the basics.
- We should be able to clone your code from Github, then run the project without doing any additional work.

Android requirements

- We encourage you to use Android Studio and Gradle.
- Do not worry about supporting old versions of Android, a minSdkVersion of 15 or later is completely fine.
- Include a pre-built APK either by email or on Github.

iOS requirements

- We encourage you to use the latest stable version of Xcode.
- Do not worry about supporting old versions of iOS, a deployment version of 10 or later is completely fine.

Priorities

1. A working app. Shortcuts are fine given the time constraints but be prepared to justify them and explain better solutions you would have implemented with more time.
2. Clean code and architecture. We would like you to write production ready code that you would be proud to submit as an open source project. While the definition of "production ready" can be subjective, the minimum requirement at Uber is to remove all the debugging code and have a reassuring amount of unit test coverage.

We expect this to take about 5 hours so no need to implement features that would obviously require more time than that. A concise and readable codebase that accomplishes all of the above requirements is the goal. Thus, do not try to do any more than is required to solve the problem cleanly. Finally, if you need more time to be proud of your delivered code, it's okay, just let us know how much time you spent on the project.

Bonus (Optional)

If you have more time to invest in this challenge and want to go further, here is a list of things we would appreciate seeing:

- **Image caching.** To save network and time, we would like you to implement a caching mechanism for the photos displayed in the app. Again, no third-party libraries can be used.
- **Basic documentation.** A README or other concise documentation that helps understand your code, design decisions, tradeoffs and things that you did not have time to implement this time, but would do so for a production app help us better understand your submission.

Good luck!

Flickr API

Try this call below:

```
https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=3e7cc266ae2b0e0d78e279ce8e361736&format=json&nojsoncallback=1&safe_search=1&text=kittens
```

It returns a JSON object with a list of Flickr photo models. The text parameter should be replaced with the query that the user enters into the app. Each Flickr photo model is defined as below:

```
{
  "id": "23451156376",
  "owner": "28017113@N08",
  "secret": "8983a8ebc7",
  "server": "578",
  "farm": 1,
  "title": "Merry Christmas!",
  "ispublic": 1,
  "isfriend": 0,
  "isfamily": 0
}
```

To load the photo, you can build the full URL following this pattern:

```
http://farm{farm}.static.flickr.com/{server}/{id}_{secret}.jpg
```

Thus, using our Flickr photo model example above, the full URL would be:

```
http://farm1.static.flickr.com/578/23451156376_8983a8ebc7.jpg
```

Notes

- More documentation about the search endpoint can be found here:
<https://www.flickr.com/services/api/explore/flickr.photos.search>
- If you have any problems with the Uber-specified API key (i.e. 3e7cc266ae2b0e0d78e279ce8e361736), feel free to generate your own here:
https://www.flickr.com/services/api/misc.api_keys.html