Project: Astronomy Picture of the Day

CS 458

1 Introduction

For this project you will create an iOS or Android app that uses the APOD API to fetch and display interesting astronomical pictures. This project is an expansion of 'Web Work' material from Week 10, and the corresponding iOS and Android demos. You are welcome to use those projects as a starting point.

When your application starts up, it should fetch the picture and associated metadata for the current day. The user should then have the option to select a different date which, if valid, will then be used to fetch a different photo. Your app will be responsible for handling user input validation, along with processing potential error cases resulting from use of the API.

The overall design of this application is largely up to you. Section 2 spells out the requirements for the interface. Section 3 covers some details around using the APOD API and requirements for how your app interacts with the network. Grading details are in Section 4, along with some extra credit opportunities, while what you need to submit is in Section 5.

2 The Main Interface

Your application must in some way display an image from the Astronomy Picture of the Day, along with the title of the image, its description, and a copyright notice, if one is provided by the API. Your interface must follow the relevant interface guidelines. Be sure the entire image fits onscreen without (significant) distortion. Scaling is fine, arbitrary stretching is not.

You must also provide a (reasonably obvious) way for the user to request a different day's image. This must allow the selection of an arbitrary date, rather than simply being 'next image' or 'previous image'. This could be part of your main interface, handled via another view/activity, or something else of your devising. You must validate the user input to guarantee that the date specified is valid and is not in the future.

3 Network and the APOD API

Your app may make use of the APOD demo API key, or you may register for your own key if you prefer. Please note that the demo key is limited to 30 requests per hour and 50 per day. Should you run up against the API limit, your app should display an appropriate notice in its main interface, rather than printing an error to the console. Other errors should also result in displaying messages in the UI (e.g. with alerts) rather than just printing console messages. Please note that the example code more or less assumes everything worked, so you will need to make some changes to support this.

Some APOD entries include videos rather than images. You are not required to implement support for these, but do need to indicate to the user that you are deliberately not showing them something, rather than having errored out or something.

Your application may not use HTTP to request resources, and must use HTTPS. If you encounter an HTTP URL, try converting it to HTTPS before running it. If you cannot fetch an image over HTTPS, display an appropriate message in some way.

4 Grading and Bonus Points

This assignment is worth a total of 50 points, broken up as follows:

10 pts	General Style / Compilation
10 pts	All required information is (nicely) displayed
15 pts	Network activity is properly structured
05 pts	App fetches/displays today's data
10 pts	User input is properly validated and pulls correct picture / information

You can earn some bonus points for going above and beyond the requirements of the assignment, to a maximum of 5 bonus points. Here are a few examples of things you might do:

- Allow the user to zoom into / scroll around in the photo (2pts)
- Support videos in some way (2pts)
- Display multiple images at a time (note: this will probably require getting your own API key) (2pts)

5 What to Submit

Zip up your project's directory, as usual. If you would like your project to be considered for extra credit, include a note with your project, or in Canvas, indicated what extra functionality is present and how to use it.