

ITIS/ITCS 6010 – Advanced Topics for Mobile App Development
In Class Assignment 5

Basic Instructions:

1. In every file submitted you **MUST** place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Student Full Name.
2. This is an individual assignment, each student is expected to work alone and submit their own work.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Create a zip file which includes all the project folder, any required libraries, and your presentation material.
6. Submission details:
 - a. You should submit the assignment through canvas: Submit the zip file.
- 7. Failure to follow the above instructions will result in point deductions.**

In Class Assignment 5 (100 Points)

In this assignment you will get familiar with iOS Concurrency and HTTP connections. You will develop an photo viewer application that downloads and displays online photos.

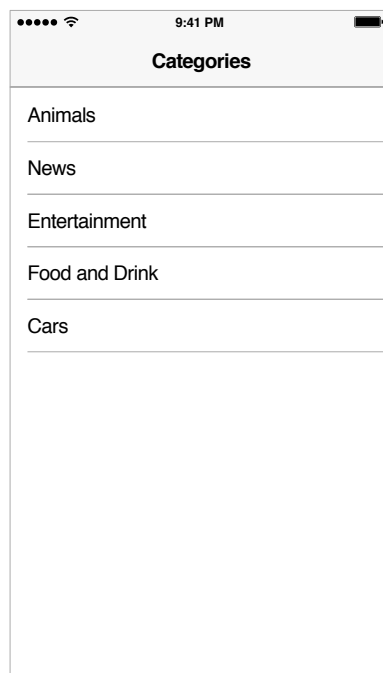
APIS:

Get photo count for category

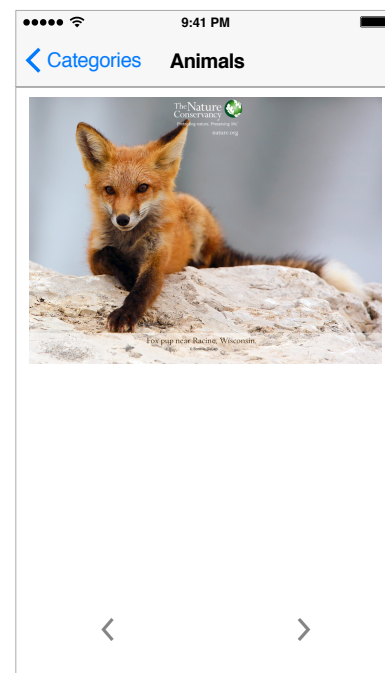
- Method: GET
- End Point: http://dev.theappsdr.com/lectures/inclass_http/photos.php
- Parameters:
 - count: should be set to get (count=get)
 - category: should be set to the requested category, the possible category values are (animals, news, entertainment, food, car)
- Response:
 - The count of the photos available in this category.

Get a photo for category

- Method: GET
- End Point: http://dev.theappsdr.com/lectures/inclass_http/photos.php
- Parameters:
 - category: should be set to the requested category, the possible category values are (animals, news, entertainment, food, car)
 - pid: is the photo index, it is from 0 to count-1.
- Response:
 - The photo at the given index.



(a) Categories



(b) Photos

Figure 1, Application Wireframe

Part 1: Categories TableView (30 Points)

As shown in Figure 1(a) this view controller should display the list of categories available.

1. When the user clicks on one of the categories the view controller should navigate to the photo view controller.
2. The selected category information should be passed to the photo view controller.

Part 2: Photo View Controller (70 Points)

As shown in Figure 1(a) this view controller should display the list of categories available.

1. When the view controller loads the “Get photo count for category” API should be called to retrieve the total number of photos for the selected category.
2. After the “Get photo count for category” API returns load the first photo for this category using the “Get a photo for category” API, which has pid=0.
3. All API calls should be performed using a child thread.
4. DO NOT store the images loaded, simply download and display the retrieved images. All UI operations should be performed by the Main Thread.
5. Upon clicking the “Next Photo” icon, you should download the next photo. For example if the currently displayed photo has pid=5 then you should download and display the photo with pid=6. If the currently displayed photo has pid=count-1, which is the last photo, you should download and retrieve the photo with pid=0.
6. Upon clicking the “Previous Photo” icon, you should download the previous photo. For example if the currently displayed photo has pid=5 then you should download and display the photo with pid=4. If the currently displayed photo has pid=0, which is the first photo, you should download and retrieve the photo with pid=count-1.
7. While the photo is being downloaded you should display an activity indicator. Also disable the next and previous buttons while the image is being downloaded. The indicator should be removed and the buttons should be activated once the image download has been completed.