AsyncTask Lab

AsyncTasks

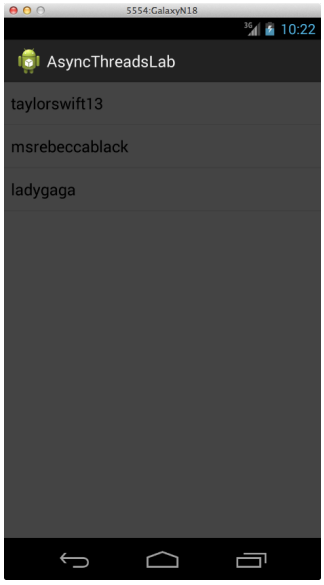
Objectives:

This week's lab aims to give you a better understanding of the AsyncTask class. Upon completing this lab, you should be able to define and use AsyncTasks to do work without blocking the application's main Thread.

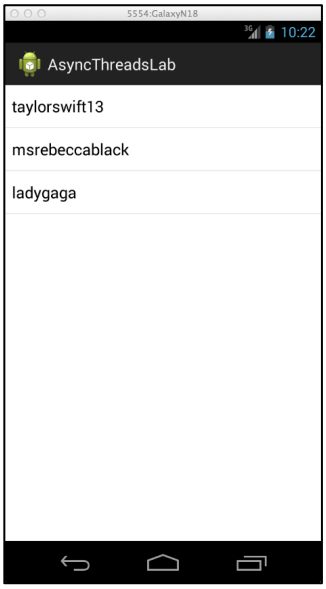
This Lab involves an app that displays simulated Twitter data. It's interface and code largely manage the selection and display of the data. You will focus on the DownloaderTaskFragment class, which is responsible for "downloading" the data (this offline version just reads data from a file) and preparing it for display.

The application comprises three Fragments: FriendsFragment, DownloaderTaskFragment and FeedFragment. The FriendsFragment displays the names of friends whose Twitter data can be displayed. The DownloaderTaskFragment starts an AsyncTask that reads the stored Twitter data and prepares it for display. The FeedFragment displays one friend's Twitter data.

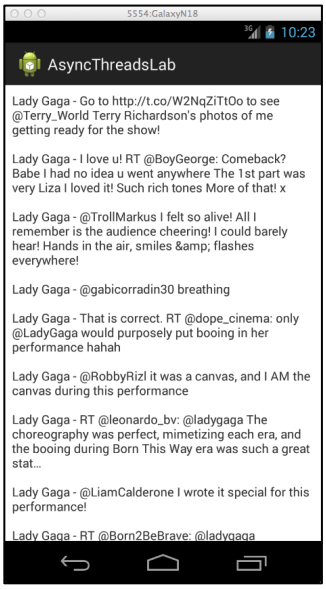
When the application's MainActivity begins running, it will set up the FriendsFragment and the DownloaderTaskFragment. At some point the user will see a ListView displaying the names of friends whose Twitter data can be displayed. Because the download process takes some time to complete, there is a period of time in which the FriendsFragment is visible, but there is no Twitter data to display. Therefore, the FriendsFragment is initially disabled, so that any user touches will be ignored. This is visible signaled to the user by greying out the FriendFragment's background.



Once the DownloaderTaskFragment has completed its work the FriendsFragment will be enabled.



After that, when a user touches a friend's name, the FeedFragment will be installed, displaying that friend's Twitter data.



The download package for this Lab contains a screencast showing the app in operation.

Implementation Notes:

1. Download the application skeleton files and import them into your IDE.
2. Implement the TODO items in DownloaderTaskFragment.java.
    a. Create a new AsyncTask subclass, called DownloaderTask, that "downloads" the Twitter data. This class should make use of the downloadTweets() helper method.
    b. Retrieve arguments from the DownloaderTaskFragment and prepare them for use with DownloaderTask.
    c. Start the DownloaderTask.
    d. Implement the DownloaderTask.

Testing:

The test case for this Lab is in the Lab5-AsyncTaskLabTest project. You can run the test case either by right clicking the project folder and then selecting Run As>Android Junit Test, or by right clicking on an individual test case class and then continuing as before. The test case is a Robotium test case. As you implement various steps of the Lab, run the test case every so often to see if you are making progress toward completion of the Lab.

Warnings:

1. These test cases have been tested on a Galaxy Nexus AVD emulator with API level 18. To limit configuration problems, you should test you app against a similar AVD. Once you've passed the test case, submit your project to Coursera.

Submission

To submit your work you will need to submit the DownloaderTaskFragment.java project file we've asked you to modify. This file should be stored in specific directories as described below and then compressed in a zip file. Then you will submit this zip file to the Coursera system. The automatic grading system will test your submission and give you feedback. This process may take some time, especially if many students are submitting at the same time. To make sure your submission is correctly graded, pay attention to the following aspects:

1. Your project files must be compressed in a zip file named AsyncTaskLabSubmit.zip.
2. When decompressed, your submission should contain one top-level directory named AsyncTaskLabSubmit. Inside that directory there should be one directory named AsyncTaskLab containing the following file: DownloaderTaskFragment.java.
3. Each new submission will overwrite any previous submissions.

Note: Once you've finished working on the Lab take a look at the code to see how it uses a "headless" Fragment (one that is not displayed) to host the AsyncTask that downloads the Twitter data. Notice how this Fragment is preserved across Activity reconfiguration, so that the AsyncTask is not destroyed and restarted in mid-operation.