

Android – Fall 2012

Assignment 4

Interacting with Web Services – Bookface client

Welcome to Bookface, a new and exciting social network. Bookface allows students to make posts and comments on a single, unified feed using their Cade login and Magic Number as their username and password. Its gonna be the next big thing.

However, as of today, Bookface doesn't have an Android app, and only has a very limited web interface. Rather than making the mobile client myself, you will create an Android app that uses Bookface's public web API to access the network.

Your app must do the following things:

1. Your code should be well organized and maintainable. I suggest you use Model View Controller.
2. Your app should have a login screen, which prompts the user for their Cade login and their Magic Number.
3. Your app should present the user with the Bookface feed, which is a list of the posts.
4. Your app should allow the user to create new posts, and edit/delete their own posts.
5. Your app should allow the user to comment on posts, and edit/delete their own comments.
6. Your app should allow the user to change their FirstName and LastName (used as a display name).
7. Every student must use their app to change their First/Last names.

You can access Bookface on the web at <http://android.eng.utah.edu>. There you will find an extremely simple (and ugly) list of the most recent posts. This is there to help you determine if your posts are actually making it to the server

You can find documentation for the webservice at <http://android.eng.utah.edu/Documentation>. The web service uses a URL based method specification. This means that the URL `"/Posts/Create"` creates a new post, and `"/Comments/Edit"` edits a comment, and so on. Parameters are passed to the web service as either a query string in the URL (for GET requests) or in the body (for POST requests). All of the methods return Json.

Full details for the methods, their parameters, and what they return can be found in the documentation.

Restrictions

As usual, no plagiarism. This includes code from the internet, as well as code samples from class. You can look at, and learn from examples found online or in class, but you **MAY NOT** copy them verbatim. The only exception to this rule is for

Gson (google's Json parsing library). If you find another parsing library you prefer, ask me and I will probably allow it (but you must ask!).

Disclaimer

Students are prohibited from posting inappropriate, disrespectful, sexual, vulgar, mean, violent, or otherwise offensive material on the feed or in comments. Doing so will result in an automatic fail of this assignment, and perhaps the class. This includes making a post and then immediately deleting it. I am backing up every single post and comment, and I will know. Don't do it. You have been warned.

Submission

You will submit a zip of your project directory via handin. This assignment will be due before Wednesday Oct 31st, at 11:59 PM.

Grading

Submissions that meet all of the requirements described above, and function without crash or error will receive a B+. To earn a grade above B+ your app should be awesome. For ideas on how to make your app awesome, consult the "Hints" section below.

Hints

1. Be careful about how often you request data from the web server. A good implementation should utilize caching to reduce data usage. Also, use the skip and take parameters when requesting posts and comments to control how much data is requested at a time.
2. Add lots of error handling. Make sure your app doesn't crash due to Internet connectivity issues or parsing errors.
3. Gracefully handle errors. Do not display technical errors or log info to the user.
4. Use good Model-View-Controller design principles.
5. Divide your application into modular Activities (posts list, create post, view comments, etc), and try to get good code reuse (some activities can be used for more than one purpose).
6. Perform all Internet communication asynchronously and on a background thread (I suggest using AsyncTask).
7. Indicate to the user when the app is loading data from the web service, and prevent them from doing anything that would be bad (like refreshing twice).
8. Look at other social networking apps like Facebook or Twitter. Consider very carefully all of the little things they do to improve the user's experience.
9. Make your app look good on phones and tablets. Use color, padding, margins, and good design to make your app enjoyable as well as functional.