Software Engineering - M1

TP 2: Android Services

J.-F. Lalande

jean-francois.lalande@insa-cvl.fr

1 Services

In this section, we focus on the use of services for achieving background tasks, for example for preparing data for the user. The idea is to regularly check new data on the server, get these data and update local data (files, database) in order to prepare it for the user. This way, the service will be available even if the smartphone goes offline.

FIGURE 1 – Flickr by Paul Downey - CC-BY

Check the following JSON API from flickr. If you look at this url at different time, you will see that the data is often updated. The idea of this section is that we will code a regular task that download the data from the JSON service and will store it in the application. This way, if the user launch the application he will be able to see the data, even if no network is available ! Thus, these JSON data will be the source of our service. The goal is to cache the data and (eventually) the images in order to have them locally.

Do not poll this url too much because we (our IP) may be banned from flickr. Thus, polling one time per minute seems a reasonable polling time.

`http://www.flickr.com/services/feeds/photos_public.gne?tags=cats&format=json`

Images can be accessed using the media tag. Different sizes are available :

`http://farm2.staticflickr.com/1441/25395426823_5ba173b194_m.jpg`
`http://farm2.staticflickr.com/1441/25395426823_5ba173b194_s.jpg`
`http://farm2.staticflickr.com/1441/25395426823_5ba173b194.jpg`

Of course we plan to cache the smallest images.

Exercise 1 Create a new Activity with 2 buttons "Start" and "Stop" that will start/stop a service.

Exercise 2 Create a service `FlickrService` that creates a timer executed at fixed rate (cf. `Timer` or `TimerTask`), for example each minute. Log some text using `Log.i("MyApp", "Test"+ time)` and check in the adb logcat that your service works fine.

Exercise 3 Perform the HTTP request against the Flickr URL : download the data from the first url (reuse the code you have used in the previous lab).

Exercise 4 Clean the JSON data. You have noted that the data are encapuslated in "jsonFlickr-Feed(...)". "jsonFlickrFeed" is not part of the JSON data and you have to remove it : use `String.subString(begin, end)` to remove the firsts and last characters.

Exercise 5 Process the JSON data : get and store the title and url of images in an object `MyFlickrData` . You can refer to the doc about JSonObject at `http://developer.android.com/reference/org/json/JSONObject.html` and this post `http://stackoverflow.com/a/9606629/1156363`.

Exercise 6 Implement a Binder `MyFlickerBinder` for your service `FlickrService`. This binder will return the object `MyFlickerData` containing the data.

Exercise 7 In your activity, create a second button "Get Data". In this new button, bind to the service using a `ServiceConnexion` in order to get back the data `MyFlickerData`. If the `Service` had enough time to run one time, you should get back some data.

Exercise 8 Display the data you got in the Activity (titles of images) in a large TextView. As the text may oversize the screen, use a ScrollView in your layout tat contains your TextView.

2 Preferences

Classical application have settings the user wish to modify. Android provide facilities for handling preferences.

Exercise 9 Customize the request to Flickr (currently asking for "cats") by adding a `PreferenceActivity` where the string can be edited

Exercise 10 Add a preference parameter for handling the request interval poll.

Exercise 11 Add a preference to handle the caching of images or not. for the ambient mode.