# IN721 Mobile Design and Development 2017 Practical 9.1 - Using Web Services

Last.fm is a music streaming site that exposes its database via a web service. Information about individual user accounts can be programmatically accessed; this requires a complex authorisation protocol that is currently out of scope for us. However, Last.FM also provides a number of public data feeds that require only an API key. In this practical, you are going to write Android apps that fetch and process JSON data from Last.fm. This handout contains enough information to get you started, but to complete the practicals, you will need to spend some time with the API documentation at http://www.last.fm/api and the AsyncTask document at developer.android.com.

### Last.fm API syntax

The root URL for the Last.fm API is http://ws.audioscrobbler.com/2.0/?

The minimum required key=value pairs are:

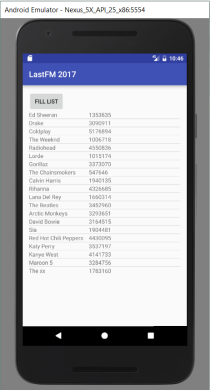
|  |  |
| --- | --- |
| **Key** | **Value description** |
| format | json |
| api\_key | Your API key value. For IN721 course work, you may use 58384a2141a4b9737eacb9d0989b8a8c |
| method | A variety of methods are available. Last.fm uses class notation for method values. For example, there is a class *chart* which exposes a method *getTopTracks*. To search on this method, add to your query string the key=value pair method=chart.getTopTracks  Some methods require you to specify a class entity. For example, the API provides the method artist.getEvents. To use this in a search, you need two key=value pairs: one for the method and one for the artist. Spaces are permitted in the artist value. For example, you can include these in your query string:  ...&method=artist.getEvents&artist=Kings of Leon |

An example complete search URL is:

### http://ws.audioscrobbler.com/2.0/?method=artist.getTopTracks&artist=Sia&api\_key=58384a2141a4b9737eacb9d0989b8a8c&limit=10&format=json

### Task 1: Display Top Artists

Use the chart.getTopArtists method to see the current top artists (as determined by some unknown algorithm). Build an Android app to display the top 20 artists, by name and with their listener count, in a ListView. You may wish to approach this task in two steps. First, simply concatenate the two data values (name and listener) into a single string and use ArrayAdapter<String> to fill the ListView. Second, implement a custom ArrayAdapter that uses a complex layout to place the data neatly. (I used a LinearLayout (horizontal) with two TextViews, and fixed the width of the first TextView.)



### Task 2: Support User Search

The Last.fm API provides a method artist.getSimilar, which returns musicians it judges to be similar to the provided artist. Implement an Android app that allows the user to type an artist’s name into an EditText, then displays 10 similar artists. To accomplish this task, you will need to go to the Android documentation and figure out the precise syntax for passing data into the doInBackground method. NB: Remember to check your returned data carefully. The JSON result for this query has a different format than that of the previous task.

### Task 3: Double Fetch and Bitmap Factory

In the JSON data returned by chart.getTopArtists are included a number of image URLs (see the JSON hierarchical display, below). Implement an Android app that displays an image of the musician currently sitting at the top of the getTopArtists list.

To accomplish this, you need to do one Http request to get the artist information (which includes the image URL), and a second Http request to get the actual image. You will also need to figure out how to convert the data you get from the InputStream into a Bitmap, which you will return from doInBackground. (Hint: Look into the class android.graphics.BitmapFactory).

