

How to get data using last.fm API?

This page documents how to use the Last.fm API in our project and how I figured out what steps I needed to follow to make that possible.

1. I started by finding the official documentation for the API, which is located here:

<http://www.last.fm/api/intro>

This page indicates right away what the URL of the web service is:

<http://ws.audioscrobbler.com/2.0/>

That means that all other methods and query requests will be appended to this base URL.

From reading this page I learned that the default response will be in XML format, which means I should specify the JSON format explicitly in the URL request, since I want my data in JSON. Finally, authentication is needed **only** if my app is going to perform write requests on a user account, which we are not planning to do.

2. Figuring out the link of the web service

I spent a few minutes looking at the different methods that the API offers: `album.getInfo`, `artist.getInfo`, `artist.search`, etc. However, all pages have this explanation:

Example URLs

To get an example URL for this API call you'll first need to [join Last.fm](#) and then [get an API account](#).

Therefore, I created an account with Last.fm, and using this login, I created an API account. To do that, there is a menu link: Add API Account, which opens a page that asks for some information. One has to fill some boxes with the application name (I made one up), URL or our webpage (I gave my CS Wellesley URL), description of app. After that, Last.fm generates an API key and Secret. Because we are creating a browser-based app, we only need the API key.

Now that I did this, whenever I go to a page of the API, I get to see example URLs for how to call a certain method:

Example URLs

JSON: /2.0/?method=album.getInfo&api_key=51de025812af79cb70f4a872936181a0&artist=Cher&al...

XML: /2.0/?method=album.getInfo&api_key=51de025812af79cb70f4a872936181a0&artist=Cher&al...

Clicking on the JSON link, I get the whole URL and I'm coloring its different parts to understand what things might change and what will be always the same.

http://ws.audioscrobbler.com/2.0/?method=album.getinfo&api_key=51de025812af79cb70f4a872936181a0&artist=Cher&album=Believe&format=json

Thus, if I'm interested in different albums of different artists, the fields artist and album will take their values with Javascript code, while I can have a fixed URL with only the API key and the format type, like below:

```
var URL =  
"http://ws.audioscrobbler.com/2.0/?api_key=51de025812af79cb70f4a872936181a0&format=json"
```

3. JSON data

The previous link returns a JSON data structure:

```
{  
  "album": {  
    "name": "Believe",  
    "artist": "Cher",  
    "id": "2026126",  
    "mbid": "86b5434d-9479-35e3-98ca-8fbcfcf4e357",  
    "url": "http://www.last.fm/music/Cher/Believe",  
    "releasedate": " 5 Jul 2005, 00:00",  
    "image": [  
      { ... }, // 2 items  
      { ... }, // 2 items  
      {  
        "#text": "http://userserve-ak.last.fm/serve/174s/88057565.png",  
        "size": "large"  
      },  
      { ... }, // 2 items  
      { ... } // 2 items  
    ],  
    "listeners": "212801",  
    "playcount": "1135885",  
  },  
}
```

It looks like a straightforward JSON object. I will be able to access the information by accessing the fields. Assuming that this JSON object was stored by a callback function in a variable response, I can access the data like this:

```
response.album.name  
response.album.playcount  
etc.
```