



## Programming assignment / Playlist messages

The task is to write a program that builds a playlist of Spotify tracks such that it looks like a message, inspired by various people on the internet doing this thing, for instance <http://spotifypoetry.tumblr.com/>.

This can be done automatically using our web API. The Python library for the web API is accessible at <http://spotipy.readthedocs.org/en/latest/>.

So, your task will be to implement a Python application doing this. You can read the input in any way you want (command line arguments, run it as a web server and take a query string, etc). Feel free to use any Python modules or frameworks that are reasonably standard.

The input to the application will be the message, and its output will be a list of Spotify tracks, i.e. if the input is "if i can't let it go out of my mind", one valid output would be:

- <http://open.spotify.com/track/6mcu7D7QuABVwUGDwovOEh>
- <http://open.spotify.com/track/5ZRxxnab9kLUqZPzoelgGP>
- <http://open.spotify.com/track/3L0bYyI0FHRiD1xZfbZedz>

(this is the beginning of one of the poems on the blog).

To be clear the output should be a playlist that reads out the original text. For instance:

- "If I Can't" by 50 Cent
- "Let It Go" by Zac Brown Band
- "Out Of My Mind" by James Blunt

And another thing to think about is that you want to break up the poem into a short playlist. I.e. it's better to turn "If I can't let it go out of my mind" into:

- If I can't
- Let it go
- Out of my mind

than to break it down into:

- If
- I can't
- Let
- It
- Go
- Out of

- My mind

Some things worth considering (though not necessarily crucial):

- Ideally, capitalization should not matter
- Should work for any UTF-8 string
- Given that the number of queries can be quite high for some sentences, can we do anything to improve the running time? Parallelization? Guarantee to return best effort result within x seconds?
- Can we re-use the results between sentences? (i.e. caching)
- What can we do to make the result as short as possible? Is there a way to find the "optimal" solution?
- What can we do about unmatched words? Can we minimize this, too? Good luck!