# Data Engineer Intern– Technical Test

**Context**

In the context of a CRM campaign, we would like to send each user the top 50 songs of their country as well as their own personal top 50 songs of the last 7 days.

For the purpose of the exercise, consider that we are receiving each day in a folder, a text file named listen-YYYYMMDD.log that contains the logs of all listening streams made on Deezer on that date. These logs are formatted as follows:

- There is a row per stream (1 listening).
- Each row is in the following format: **sng_id|user_id|country**

With:

- **sng_id**: Unique song identifier, an integer. For your information, Deezer catalog contains more than 80M songs, a number that is constantly increasing.
- **user_id**: Unique user identifier, an integer. Deezer currently has millions of users, a number that is constantly increasing.
- **country**: 2 characters string upper case that matches the country ISO code (Ex: FR, GB, BE, ...). There are 249 existing country codes, this number rarely changes (only when there is massive geopolitical change).

In the context of the exercise, we are considering that the daily number of streams is around 30M. We should expect that the file contains occasionally corrupted rows that do not respect the format given above. Therefore, having some corrupted rows as input should not have an impact on the proper functioning of the script.

As an example, a sample of the data is provided in a separate email.

## Objective

The objective of this exercise is to suggest a system that computes on a daily basis, the top 50 songs the most listened in each country on the last 7 days, as well as the top 50 songs (optional) the most listened by each user on the last 7 days.

To reach this goal, you will have to suggest a set of scripts that produces each day 1 text file:

**1) [Must have] country_top50_YYYYMMDD.txt** on which each row contains the top 50 songs listened in a specific country, on the specified format:

**country|sng_id1:n1,sng_id2:n2,sng_id3:n3,...,sng_id50:n50**

where *country* is the country ISO code (2 characters long)
*sng_id1:n1* the identifier of the song the most listened with the related number of streams
*sng_id2:n2* the identifier of the 2nd song the most listened with the related number of streams
etc...

## Constraints

To solve this exercise, you can either use Python / Scala / Java and its libraries. You are allowed to write on disk as many intermediary files as you want to, of the format of your choice. However, you **will not use any third-party system** that will require to run a service (no SGBD mysql, posgres, mongodb, etc..., no Hadoop, MapReduce, etc...).

Your solution should preferably use few RAM (less than 1 Go) – you have the ability to write intermediate information on the disk in necessary. The suggested system can save on disk intermediate information that will be reused the following day.

The solution should run on Linux, being easily readable and maintainable.

## Expected delivery

A ZIP archive of your project including

- as many files as needed (scripts, programs, functions, etc…)

- a README file describing the solution implemented, how to run it every day to compute the files.

Your code should be

- **Maintainable**: simple, clean and easy to understand.

- **Performant**: it must be production-ready. Consider the complexity of your algorithm.