



# Générateur de playlit PlaylistGen

Cahier des charges fonctionnel

## Table des matières

<b>1</b>	<b>Présentation du problème</b>	<b>2</b>
1.1	Contexte . . . . .	2
1.2	Projet . . . . .	2
1.2.1	Listes exhaustives des éléments et contraintes . . . . .	2

# 1 Présentation du problème

## 1.1 Contexte

La section BTS SIO du Lycée André Malraux gère, maintient et anime la Radio Libre de l'établissement.

La nature des morceaux qui sont écoutés par les auditeurs provient du travail du programmeur radio dont la responsabilité est de concevoir l'enchaînement musical diffusé.

Pour cela, le programmeur radio utilise le logiciel *VLC*. Il ajoute à la main tous les morceaux qu'il souhaite diffuser. Une fois ajoutés dans la liste de lecture de *VLC*, le programmeur radio ordonne les morceaux selon la dynamique qu'il souhaite donner, et enfin il sauvegarde cette liste de lecture (*playlist*) dans le format *M3U* et le stocke dans le système de fichier à l'endroit défini par l'équipe de la radio.

*Tout cela est bien fastidieux*, se dit le programmeur radio.

Non content de ce dur labeur, le programmeur radio en appelle aux compétences d'un binôme de développeurs pour lui offrir un outil de génération automatique de playlist.

## 1.2 Projet

L'objectif global est de fournir à l'organisation un outil de génération de playlist musicale au format initial en *M3U* et *XSPF* afin de correspondre aux attentes techniques de *VLC*.

Par ailleurs, l'avantage de ces formats de playlist est qu'ils sont standardisés et sont ainsi lisibles depuis la majorité des lecteurs multimédia.

### 1.2.1 Listes exhaustives des éléments et contraintes

L'environnement de fonctionnement est un système d'exploitation GNU/Linux dont l'interface graphique est extrêmement peu utilisée.

La majorité des applications et programmes employés le sont en interface de ligne de commande 3.

La seule chose dont nous disposons est une base de données des morceaux musicaux regroupant les champs suivants : *Id*, *Titre*, *Album*, *Artiste*, *Genre*, *Sous-Genre*, *Durée*, *Polyphonie*, *Format*, *Chemin*.

Cette base de données utilise le format de stockage SQLite et est associée au programme dans un fichier nommé : `musiqueDB.sqlite`.

La structure et le jeu de données de la base sont disponibles au téléchargement sur : <https://github.com/btsmalrauxallennes/generateur-de-playlist>.

Vous trouverez dans le 1 page suivante la structure de la base de données au format SQL/92.

L'organisation utilise principalement des outils libres et souhaite que ses développements soient libres eux aussi.

Listing 1 – Script SQL de la structure de la base de données employée

```
1 SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
2 SET time_zone="+00:00";
3
4 /* !40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
5 /* !40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
6 /* !40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
7 /* !40101 SET NAMES utf8 */;
8
9 DROP TABLE IF EXISTS musique;
10 CREATE TABLE IF NOT EXISTS musique (
11     id int (10) unsigned NOT NULL AUTO_INCREMENT,
12     titre varchar (255) COLLATE utf8_unicode_ci NOT NULL,
13     album varchar (255) COLLATE utf8_unicode_ci NOT NULL,
14     artiste varchar (255) COLLATE utf8_unicode_ci NOT NULL,
15     genre varchar (255) COLLATE utf8_unicode_ci NOT NULL,
16     sousgenre varchar (255) COLLATE utf8_unicode_ci NOT NULL,
17     duree int (11) NOT NULL,
18     format varchar (255) COLLATE utf8_unicode_ci NOT NULL,
19     polyphonie tinyint (4) NOT NULL DEFAULT 2,
20     chemin varchar (255) COLLATE utf8_unicode_ci NOT NULL,
21     PRIMARY KEY (id)
22 ) ENGINE = MyISAM DEFAULT CHARSET = utf8 COLLATE = utf8_unicode_ci
    AUTO_INCREMENT=1403;
```