-Créer une connexion/obtenir autorisation API Spotify :

* <https://github.com/jwilsson/spotify-web-api-php/blob/master/docs/examples/access-token-with-authorization-code-flow.md>
* https://gist.github.com/jwilsson/07f30721b5a3378c78bf4c86cdc92b98

**Pour la mise en place des requêtes en PHP s’inspirer de :**

* <https://github.com/jwilsson/spotify-web-api-php/blob/master/src/SpotifyWebAPI.php>

-GetRandomSong (Récupération de son aléatoire dans la librairie spotify) :

* https://developer.spotify.com/documentation/web-api/reference/search/search/

function getRandomSearch() {

// Liste des caractères pouvant être choisis aléatoirement pour la recherche

const characters = 'abcdefghijklmnopqrstuvwxyz';

// Récupération d’un caractère aléatoire dans characters

const randomCharacter = characters.charAt(Math.floor(Math.random() \* characters.length));

let randomSearch = '';

// On place le caractère choisis au début, ou à l’intérieur de la recherche, aléatoirement

switch (Math.round(Math.random())) {

case 0:

randomSearch = randomCharacter + '%';

break;

case 1:

randomSearch = '%' + randomCharacter + '%';

break;

}

return randomSearch;

}

//On récupère un index aléatoire

const randomOffset = Math.floor(Math.random() \* 10000);

//Requête avec les paramètres aléatoires et le type track (on récupère que des titres, et non des albums etc..)

GET https://api.spotify.com/v1/search?

type=track

&q=getRandomSearch()

&offset=randomOffset

-GetTrackGenre (Récupération du genre d’un titre : !on ne peut pas le récupérer directement depuis les infos d’un titre) :

On récupère les infos d’un « track », et on stocke l’URI de l’album.

* <https://waytolearnx.com/2019/10/lire-un-fichier-json-avec-php.html>

A partir de l’URI, on récupère l’ID de l’album (on enlève « spotify:album: ») et on fait une requête pour obtenir les infos de l’album.

* <https://developer.spotify.com/documentation/web-api/reference/albums/get-album/>

GET [https://api.spotify.com/v1/albums/{id}](https://api.spotify.com/v1/albums/%7bid%7d)

On, decode le JSON, et on récupère le tableau « genres ».

-GetRandomSongByGenre (Récupération d’un titre aléatoire selon un genre sélectionné) :

* https://developer.spotify.com/documentation/web-api/reference/search/search/#writing-a-query---guidelines

GET https://api.spotify.com/v1/search?

type=track

&q=getRandomSearch()%20genre:{genreName}

&offset=randomOffset

-GetUserPlaylists (On récupère les playlists de l’utilisateur pour vérifier si la playlist « Nomdelapp » existe) :

* https://developer.spotify.com/documentation/web-api/reference/playlists/get-a-list-of-current-users-playlists/

GET https://api.spotify.com/v1/me/playlists

-CreatePlaylist (On créé une playlist “Nomdelapp” si elle n’existe pas déjà) :

* https://developer.spotify.com/documentation/web-api/reference/playlists/create-playlist/

POST [https://api.spotify.com/v1/users/{user\_id}/playlists](https://api.spotify.com/v1/users/%7buser_id%7d/playlists)

JSON DATA "name": “Nomdelapp”, "public": false

-AddTrackToPlaylist (On ajoute un titre à la playlist « Nomdelapp ») :

* https://developer.spotify.com/documentation/web-api/reference/playlists/add-tracks-to-playlist/

POST [https://api.spotify.com/v1/playlists/{playlist\_id}/tracks?uris=spotify%3Atrack%{track\_id}](https://api.spotify.com/v1/playlists/%7bplaylist_id%7d/tracks?uris=spotify%3Atrack%25%7btrack_id%7d)

//Idem pour DELETE

|  |  |
| --- | --- |
| pop | k-pop, pop-film, j-pop, indie-pop, synth-pop |
| monde | Brazil, british, french, german, indian, iranian, philippines-opm, Spanish, Swedish, Turkish, world-music |

Listes des playlists : "genres": [

* + "acoustic",
  + "afrobeat",
  + "alt-rock",
  + "alternative",
  + "ambient",
  + "anime",
  + "black-metal",
  + "bluegrass",
  + "blues",
  + "bossanova",
  + "breakbeat",
  + "cantopop",
  + "chicago-house",
  + "children",
  + "chill",
  + "classical",
  + "club",
  + "comedy",
  + "country",
  + "dance",
  + "dancehall",
  + "death-metal",
  + "deep-house",
  + "detroit-techno",
  + "disco",
  + "disney",
  + "drum-and-bass",
  + "dub",
  + "dubstep",
  + "edm",
  + "electro",
  + "electronic",
  + "emo",
  + "folk",
  + "forro",
  + "funk",
  + "garage",
  + "gospel",
  + "goth",
  + "grindcore",
  + "groove",
  + "grunge",
  + "guitar",
  + "happy",
  + "hard-rock",
  + "hardcore",
  + "hardstyle",
  + "heavy-metal",
  + "hip-hop",
  + "holidays",
  + "honky-tonk",
  + "house",
  + "idm",
  + "indie",
  + "industrial",
  + "j-dance",
  + "j-idol",
  + "j-rock",
  + "jazz",
  + "kids",
  + "latin",
  + "latino",
  + "malay",
  + "mandopop",
  + "metal",
  + "metal-misc",
  + "metalcore",
  + "minimal-techno",
  + "movies",
  + "mpb",
  + "new-age",
  + "new-release",
  + "opera",
  + "pagode",
  + "party",
  + "piano",
  + "post-dubstep",
  + "power-pop",
  + "progressive-house",
  + "psych-rock",
  + "punk",
  + "punk-rock",
  + "r-n-b",
  + "rainy-day",
  + "reggae",
  + "reggaeton",
  + "road-trip",
  + "rock",
  + "rock-n-roll",
  + "rockabilly",
  + "romance",
  + "sad",
  + "salsa",
  + "samba",
  + "sertanejo",
  + "show-tunes",
  + "singer-songwriter",
  + "ska",
  + "sleep",
  + "songwriter",
  + "soul",
  + "soundtracks",
  + "study",
  + "summer",
  + "tango",
  + "techno",
  + "trance",
  + "trip-hop",
  + "work-out",
  + "world-music"

]

}