

# Documentation : Spotify MP3 Sync

Auteur : Noa Sitbon

Version : 1.0

Date : décembre 2025

Ce document fournit les instructions détaillées pour l'installation, la configuration et l'utilisation de l'outil Python **Spotify MP3 Sync**, qui maintient votre bibliothèque de titres likés Spotify synchronisée avec un dossier local de fichiers MP3.

## I. Prérequis du Système

Pour garantir le bon fonctionnement de l'application, les éléments suivants sont nécessaires :

### 1. Environnement Logiciel

Logiciel	Version Minimale	Rôle
Python	3.8	Langage d'exécution du script.
FFmpeg	Dernière stable	Outil externe crucial pour la conversion des fichiers audio en MP3 (utilisé par yt-dlp).

### 2. Configuration FFmpeg

FFmpeg doit être accessible par votre système.

Téléchargez FFmpeg et ajoutez le répertoire bin contenant l'exécutable (ffmpeg.exe, ffmpeg) à la variable d'environnement PATH de votre système d'exploitation.

## II. Configuration de l'API Spotify

Pour interagir avec votre compte Spotify, vous devez créer une application développeur et obtenir des identifiants (Client ID et Secret).

### 1. Création de l'Application

- Accédez au **[Spotify Developer Dashboard]** (<https://developer.spotify.com/dashboard/applications>).
- Cliquez sur le bouton **"Create an app"**.

3. Fournissez un nom et une description (ex: Ma Sync MP3).

## 2. Obtention des Identifiants

Une fois l'application créée :

1. Notez le **Client ID**.
2. Cliquez sur "**Show Client Secret**" et notez la valeur.

## 3. Définition de l'URI de Redirection (Callback)

L'API Spotify a besoin d'une adresse de retour pour l'authentification :

1. Cliquez sur "**Edit Settings**" dans votre application.
2. Dans le champ "**Redirect URIs**", ajoutez l'URL suivante :  
`http://localhost:8080`
3. Cliquez sur "**Save**".

# III. Installation et Configuration du Projet

## 1. Téléchargement du Projet

Assurez-vous d'avoir le dossier du projet contenant tous les fichiers Python (main.py, downloader.py, etc.).

## 2. Installation des Dépendances Python

Ouvrez une invite de commande (terminal) dans le répertoire racine du projet et exécutez la commande suivante pour installer les bibliothèques requises (spotipy, yt-dlp, python-dotenv) :

```
pip install spotipy yt-dlp python-dotenv
```

## 3. Création du Fichier d'Environnement

Créez un fichier nommé **.env** à la racine de votre projet. Ce fichier stocke vos identifiants de manière sécurisée et ne doit **jamais être partagé**.

**Contenu du fichier .env :**

```
SPOTIPY_CLIENT_ID="VOTRE_CLIENT_ID_QUE_VOUS_AVEZ_NOTE"  
SPOTIPY_CLIENT_SECRET="VOTRE_CLIENT_SECRET_QUE_VOUS_AVEZ_NOTE"  
SPOTIPY_REDIRECT_URI="http://localhost:8080"
```

# IV. Utilisation et Synchronisation

Le script principal main.py gère l'ensemble du processus de synchronisation.

## 1. Première Exécution et Authentification

Lors de la toute première exécution, le script vous guidera à travers le processus d'autorisation :

1. Exécutez le script :  
`python main.py`
2. Le script ouvrira automatiquement votre navigateur sur une page de connexion Spotify.
3. Connectez-vous à votre compte et **autorisez** l'application à lire votre bibliothèque.
4. Une fois l'autorisation accordée, le navigateur tentera d'ouvrir `http://localhost:8080`. Le script Python interceptera le code d'authentification et sauvegardera votre jeton d'accès dans un fichier caché pour les utilisations futures.

## 2. Le Cycle de Synchronisation

Chaque fois que vous exécutez `python main.py`, le processus de synchronisation se déroule en trois phases :

Phase	Description	Fichiers Impliqués
I. Récupération	Le script récupère la liste actuelle de <b>tous</b> vos titres likés sur Spotify (via <code>spotify_liked_tracks.py</code> ).	<code>spotify_auth.py</code> , <code>spotify_liked_tracks.py</code>
II. Nettoyage Local	Le script compare les titres likés actuels avec sa base de données ( <code>tracks.db</code> ). Si un titre est dans la base de données mais <b>n'est plus liké</b> , il est <b>supprimé</b> du dossier <code>mp3/</code> et de la base de données.	<code>mp3_manager.py</code>
III. Téléchargement	Le script vérifie les titres likés qui <b>ne sont pas</b> dans la base de données. Il recherche le morceau sur YouTube et le télécharge au format MP3 dans le dossier <code>mp3/</code> .	<code>downloader.py</code>

### 3. Résultat

Après la synchronisation, vous trouverez :

- **Dossier mp3/** : Contient tous les fichiers MP3 synchronisés, nommés selon le format : Titre du Morceau - Nom de l'Artiste.mp3.
- **Fichier tracks.db** : Base de données SQLite utilisée par mp3\_manager.py pour suivre quels titres ont déjà été téléchargés (empêchant les téléchargements répétés).

## V. Structure des Fichiers du Projet

Voici l'organisation des fichiers et leur rôle dans le projet :

Spotify-MP3-Sync/

— main.py	# POINT D'ENTRÉE : Orchestre toutes les étapes de la synchronisation.
— mp3_manager.py	# Gère la base de données SQLite et la suppression des fichiers.
— downloader.py	# Gère l'API yt-dlp pour le téléchargement/conversion en MP3.
— spotify_auth.py	# Gère l'authentification OAuth avec Spotify et l'objet client
— spotify_liked_tracks.py	# Fonctions pour la récupération paginée des titres likés.
— .env	# Contient les identifiants Client ID/Secret. (À NE PAS PARTAGER)
— mp3/	# Dossier où les fichiers MP3 sont enregistrés.
— tracks.db	# Base de données de suivi SQLite (créée automatiquement).