**Step-by-Step Guide to Set Up the Automated YouTube Playlist Creator**

This guide walks you through all the steps required to set up the application from scratch and start using it with your own Google account.

**1. Set Up Google Cloud Console for YouTube Data API**

**Step 1: Go to the Google Cloud Console**

* Visit: <https://console.cloud.google.com/>
* Sign in with your Google account.

**Step 2: Create a New Project**

* Click the project dropdown in the top navbar.
* Click **"New Project"**.
* Give your project a name (e.g., YouTubePlaylistCreator) and click **"Create"**.

**Step 3: Enable YouTube Data API v3**

* In the left sidebar, go to **APIs & Services > Library**.
* Search for **YouTube Data API v3**.
* Click on it, then click **"Enable"**.

**Step 4: Create OAuth Credentials**

* Go to **APIs & Services > Credentials** in the sidebar.
* Click **"+ Create Credentials" > "OAuth client ID"**.
* If prompted, configure the OAuth consent screen (select **External**, then add basic info).
* Under "Application type," choose **Desktop app**.
* Name it (e.g., YTPlaylistApp) and click **"Create"**.

**Step 5: Download Credentials File**

* After creation, click the **"Download JSON"** icon next to your new client ID.
* This will download a file named something like client\_secret\_xxx.json.
* Rename this file to credentials.json and replace it in the root directory of the project folder.

**2. Install Python and Dependencies**

**Step 6: Install Python**

Make sure you have Python 3.7+ installed:

python --version

If not installed, download it from <https://www.python.org/downloads/>.

**Step 7: Create Virtual Environment (optional but recommended)**

python -m venv venv

source venv/bin/activate # Windows: venv\Scripts\activate

**Step 8: Install Required Packages**

pip install -r requirements.txt

**3. Prepare Your Song Files**

**Step 9: Add Songs to songs/ Directory**

* Create a folder named songs in the project directory (if not already present).
* Copy your .mp3, .wav, or .flac audio files into this folder.
* Format filenames as Artist - Song Title.mp3 for best search results.

**4. Run the Script and Authorize**

**Step 10: Run the Script**

python playlist\_creator.py

**Step 11: Log In and Authorize**

* A browser window will open.
* Select your Google account and allow the required permissions.
* After successful login, the script will begin reading song names and creating your playlist.

**5. Tips and Best Practices**

* If you see API quota errors, you may need to request more quota in the Google Cloud Console.
* Try to keep song names clean and standardized for better search accuracy.
* Each OAuth authorization creates a token.json that stores your session—delete it to force re-auth.

You are now ready to use the **Automated YouTube Playlist Creator**!