**Step 1: Python VS Code setup**

1. Install python 3.9 + (3.14)
2. Install VS code
3. Enable Python extension in VS code

 **Open VS Code**

* File → Open Folder → select your project folder (mcp-youtube-poc).
* Save server.py

 **Open a terminal inside VS Code**

* View → Terminal → cmd or PowerShell.
* In powerShell >

> D:\Bos\_Framework\MCP\mcp\_youtube\_poc> python server.py

* This runs the python code
* Note: CTRL + C to stop
* To get into shell

> D:\Bos\_Framework\MCP\mcp\_youtube\_poc> python

 **Set up virtual environment (keeps dependencies clean):**

**Step 2: Google Cloud**

**Detailed Step-by-step — Create a Google Cloud Project & enable YouTube Data API (POC)**

Below are the exact UI steps + small commands you can follow in one sitting. Estimated time: **15–30 minutes**.

**Prerequisites**

* A Google account (the one you used for the Google Cloud welcome screen).
* Optional: your free $300 trial/billing account ready (not required for a basic API key but useful if you plan more calls).

**1) Create a new Google Cloud Project**

1. Open **Google Cloud Console**: https://console.cloud.google.com
2. At the top bar, click the **Project dropdown** (it shows current project name) → **New Project**.
3. In the **Create project** dialog:
   * **Project name**: e.g. mcp-youtube-poc
   * **Organization / Location**: leave default if not required.
   * Click **Create**.
4. Wait a few seconds; the new project will be created. Make sure it is **selected** (project name visible in top bar).

**Output:** Project ID and Project Number (you’ll need Project ID if using gcloud later).

**2) (Optional) Link a billing account**

* If you want more quota or plan bigger testing, go to **Billing → Link a billing account** and follow the prompts.
* For simple public-data calls with an API key you often do **not** need billing.

**3) Enable the YouTube Data API v3**

1. In the left nav, go to **APIs & Services → Library**.
2. In the Library search box, type **YouTube Data API v3** and open it.
3. Click **Enable**.

**Output:** The API is enabled for your project.

**4) Create an API key (fastest for public channel data)**

1. In **APIs & Services** left menu → **Credentials**.
2. Click **+ CREATE CREDENTIALS → API key**.
3. A new key appears. **Copy** it (you’ll use it in .env).

**API Key Example:** AIzaSyDZcslpnx8EwDII\_Cu2Nag3d6jkL3\_xnjw

**Restrict the key (very important for security)**

1. Click **Restrict key** (or edit the key right away):
   * **Application restrictions**: choose one appropriate for your setup:
     + None (quick dev — not recommended long-term), or
     + IP addresses (if calling from a fixed server IP), or
     + HTTP referrers (if from a web app).
   * **API restrictions**: select **Restrict key** → pick **YouTube Data API v3**.
2. Save the restrictions.

**Why restrict?** Unrestricted keys can be abused if leaked. For local dev you can leave less restricted but tighten before any publish.

**5) (Optional) OAuth 2.0 Client ID — if you need private channel/owner data**

* If you must access a channel’s private data (e.g., manage channel, upload, access non-public analytics), create an **OAuth Client ID** instead:
  1. **OAuth consent screen** → configure (App name, email, test users).
  2. Then **Credentials → Create Credentials → OAuth client ID** → choose **Desktop app** for local dev.
  3. Download the JSON and use OAuth flow in your app.
* For simple public metrics (views, subscribers) an **API key** is sufficient.

**Step 3: Setup**

* 1. Create a text file “requirement” without .txt at the end in VS code it contains all the necessary installs for python:

mcp

google-api-python-client

python-dotenv

faiss-cpu

sentence-transformers

youtube-transcript-api>=0.6.2

numpy

Run this in VS code terminal as

Powershell

*> D:\Bos\_Framework\MCP\mcp\_youtube\_poc> pip install -r requirements.txt*

* 1. Create the youtube API file for linking as “.env” file for API Key

 In VS Code, **Right-click** your project folder → **New File**.

 Name it:

.env

(yes, just .env, no extension like .txt).

 Inside that file, paste your YouTube API key like this:

YOUTUBE\_API\_KEY=AIza...your\_key\_here...

 Save the file.

Test with this code by creating test.py

from dotenv import load\_dotenv

import os

# Load environment variables

load\_dotenv()

# Test: print the API key

print("Loaded API key:", os.getenv("YOUTUBE\_API\_KEY"))

run test.py should show the API key

*> D:\Bos\_Framework\MCP\mcp\_youtube\_poc> python test.py*

* 1. Creating the server\_2.py code which has all MCP tools



Test the code by running in terminal powershell

*python server\_2.py*

If you get messages like server active the code runs without error

* 1. Create a testing code to test the MCP is Server.py code



Run it again in powershell

Python test\_transcript.py server

* 1. Download and Install Claude desktop

[Download Claude](https://claude.ai/download)

* 1. Create a laude\_desktop\_config.json in VS code



*{*

*"mcpServers": {*

*"youtube": {*

*"command": "python",*

*"args": ["D:/Bos\_Framework/MCP/mcp\_youtube\_poc/server.py"],*

*"env": {*

*"PYTHONUNBUFFERED": "1"*

*}*

*}*

*}*

*}*

**Note: server.py should be updated to your MCP tool code**

Location of Claude Config

On Windows, the Claude Desktop config lives here:

Win+R

*%APPDATA%\Claude\*

Enter

Here paste the claude\_desktop\_config.json

* 1. Now open claude desktop it should automatically connect to Youtube MCP server

Ask prompt “ what MCP servers are you connected to?”

If successfully connected it should give MCP youtube along with web MCP

* 1. ALL GOOD to go, experiment and have fun