How to Create a Local YouTube Front-End, Quick & Dirty

written by Traveler

You need:

- Gmail account
- API key for YouTube Data API v3
- Python
- Linux (though it could, in theory, work with Windows too)
- Cloned the repository from https://github.com/EdgeOfAssembly/LocalTube into LocalTube directory somewhere (I presume you have already done it, ottherwise you would not be reading this. Duh!)

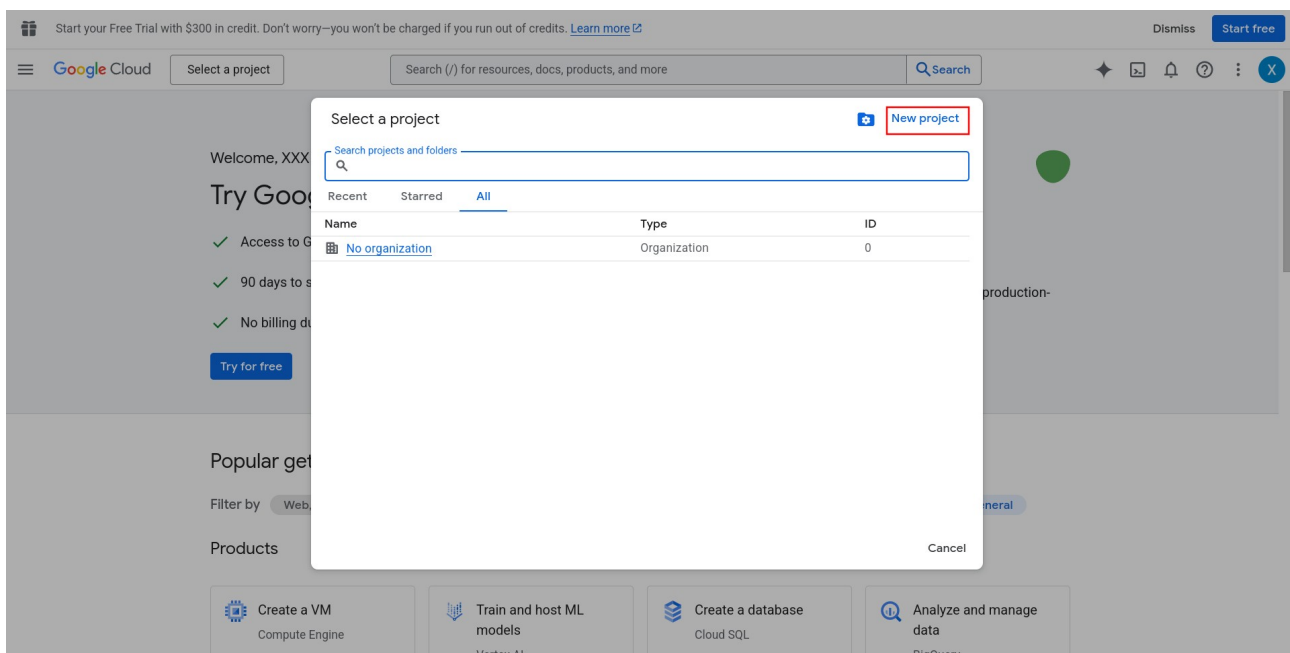First, I assume you already have a Gmail account created.
The next step is to create a project (because we can't have an API key without a project).
Go to: console.cloud.google.com and log in with your Gmail account email and password.
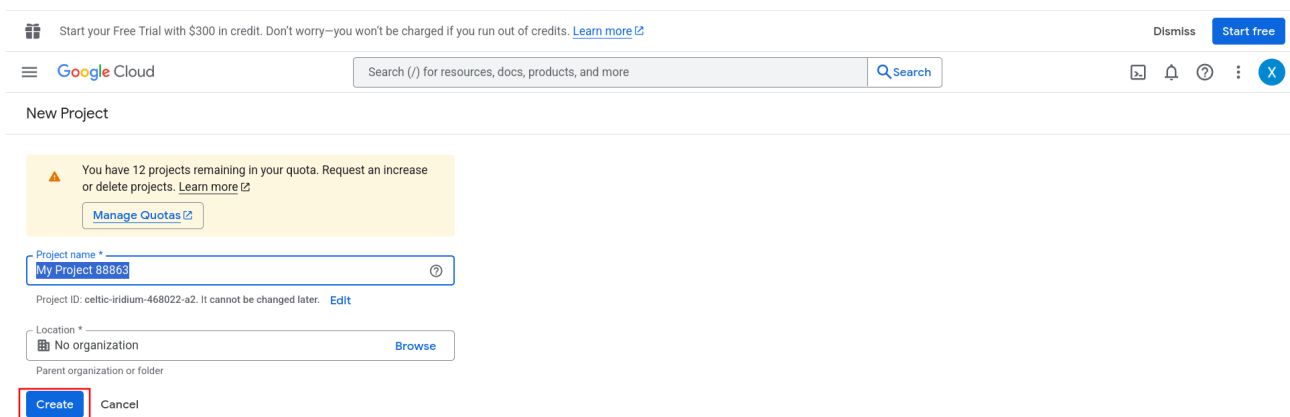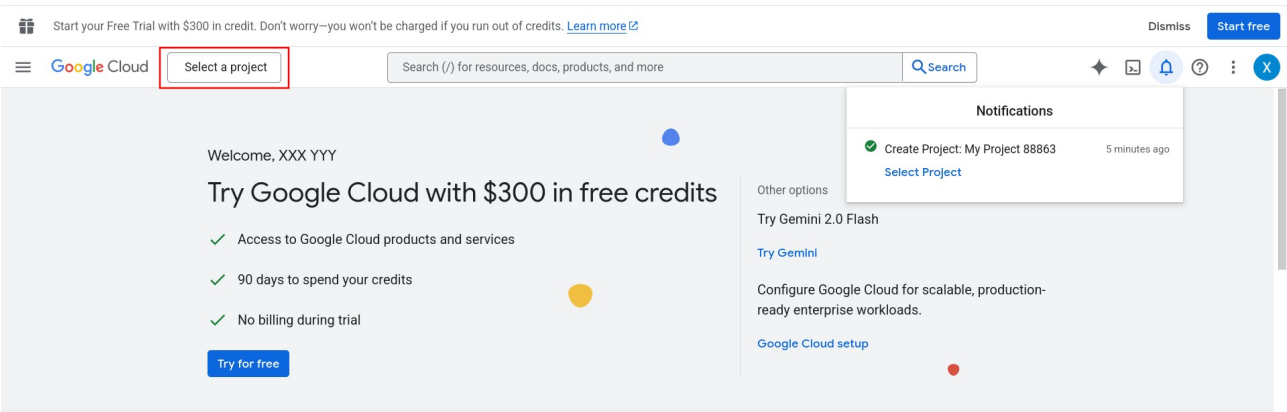You should see something like this after you agree to the Terms of Service. Click "Select a project."
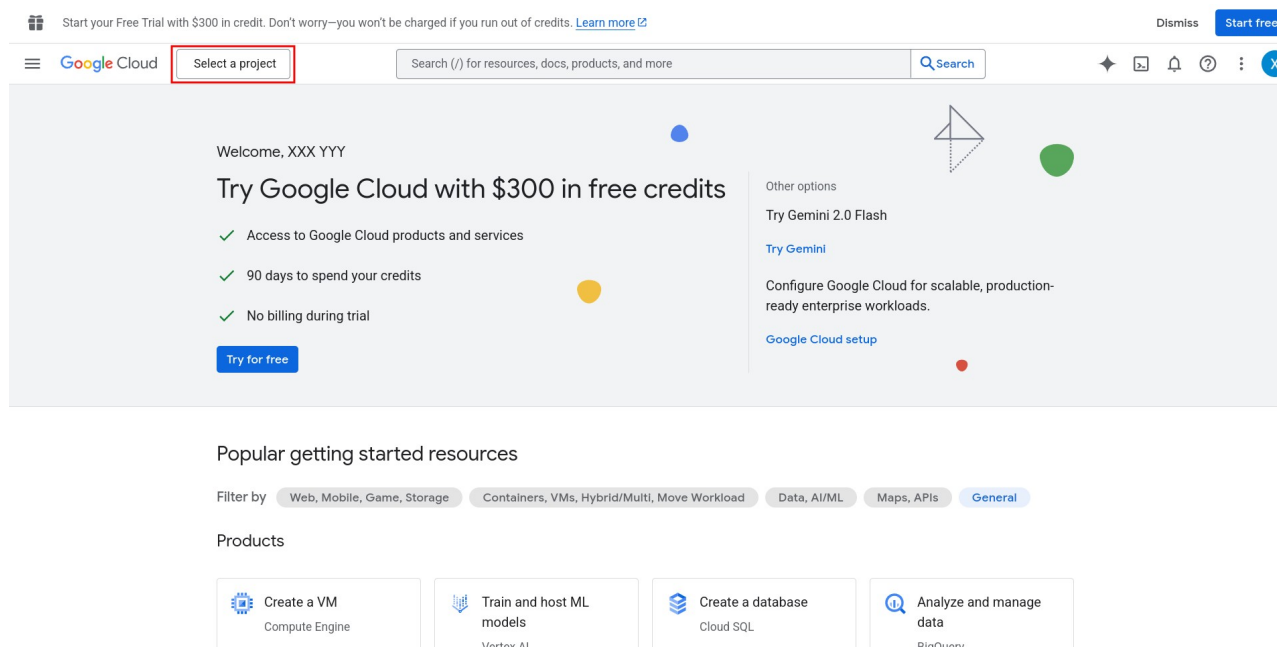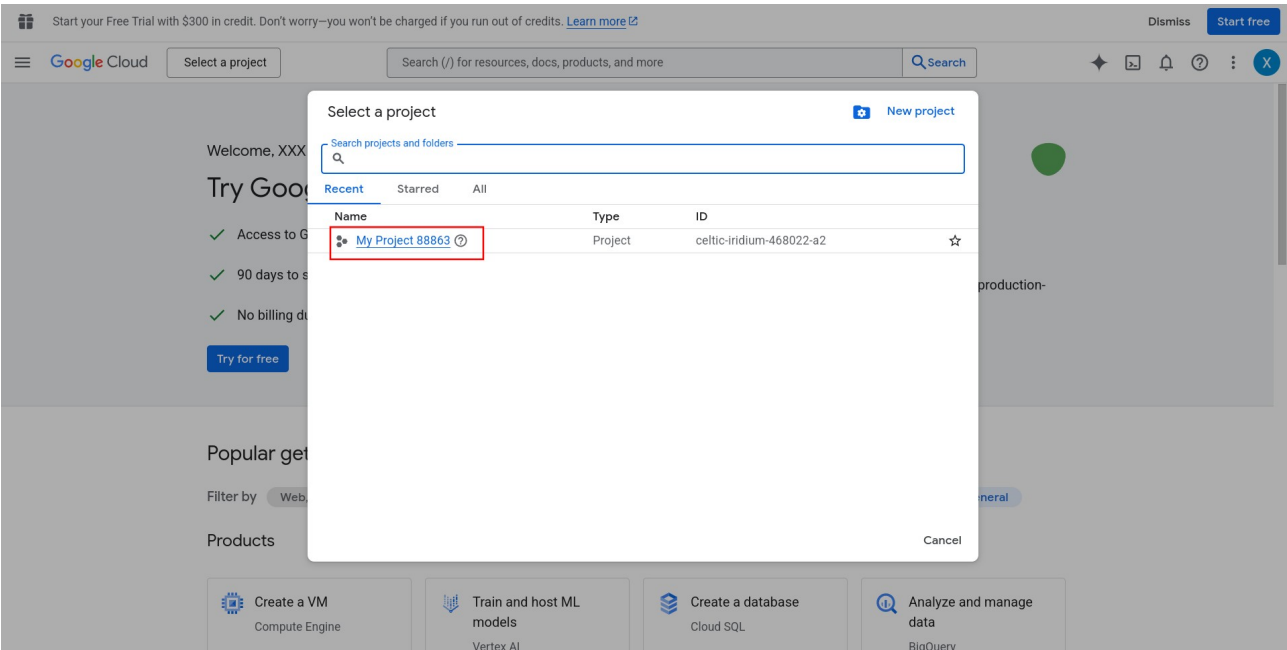
Click "New project."

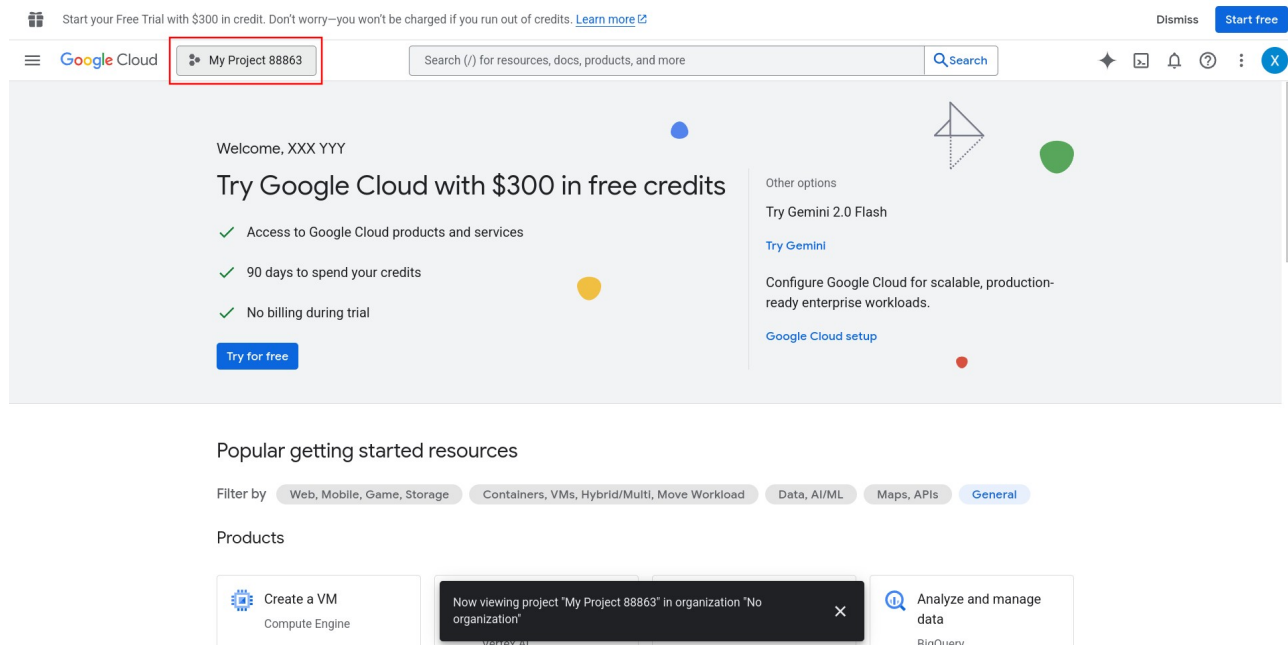Use the given project name or enter anything you want, then click "Create."
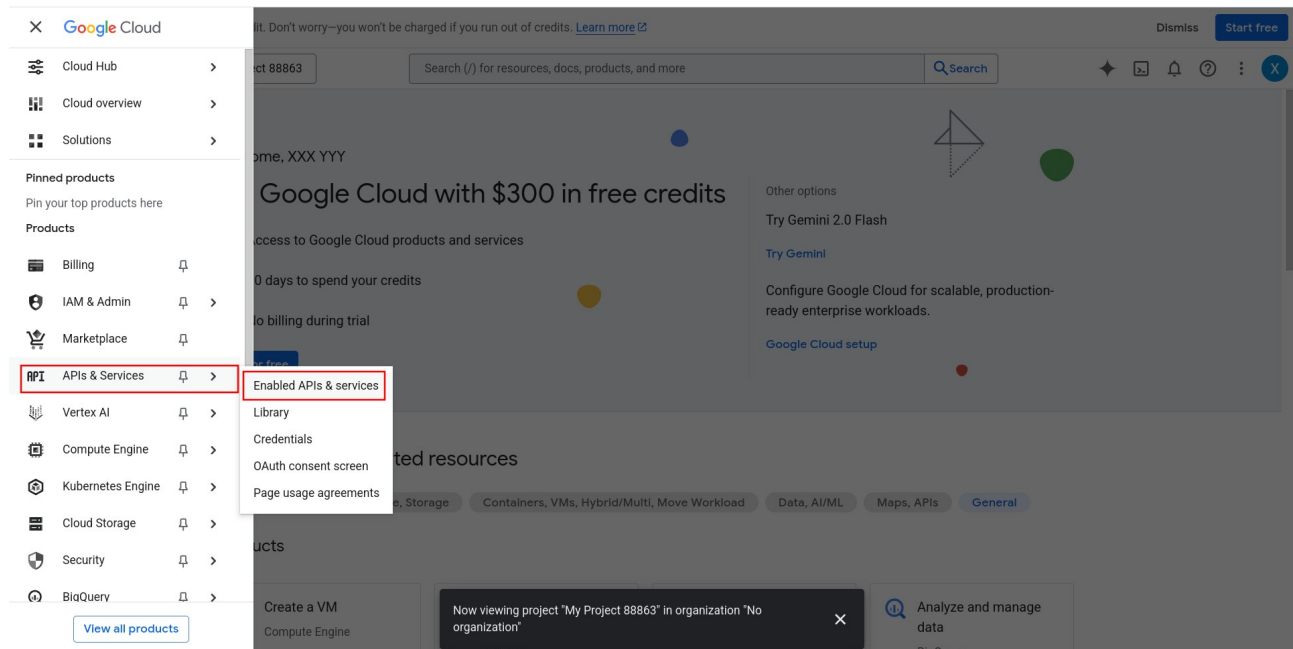
Click "Select a Project" again.
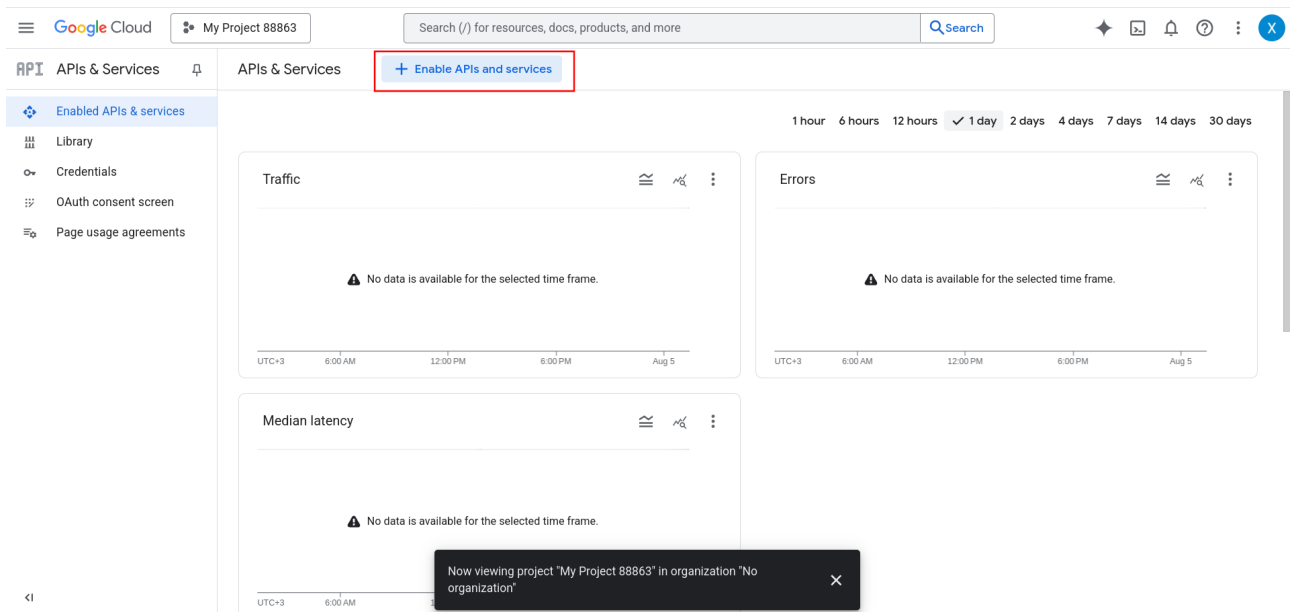

Click your project name.
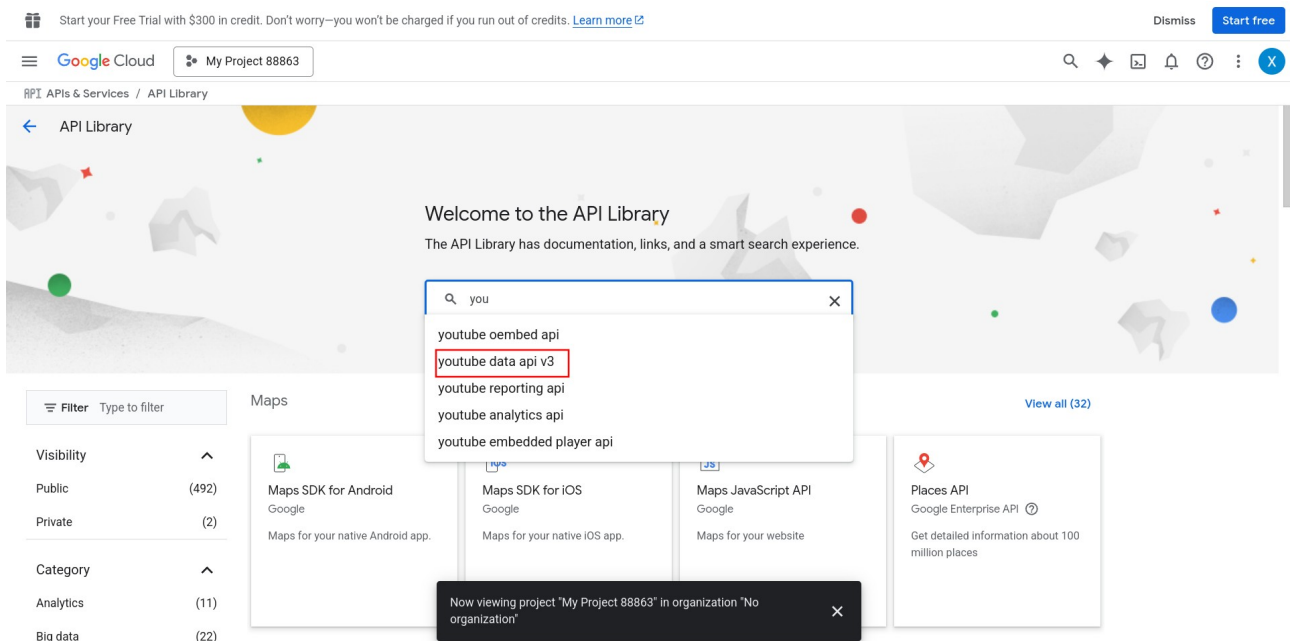
Your project is now selected.

Next, you need to enable the YouTube Data API v3 and get an API key.
From the left, open the menu, select "APIs & Services," and then click "Enable APIs & Services."

Then click "Enable APIs and Services" again.

In the search bar, type "you" and select "YouTube Data API v3."

Click the "YouTube Data API v3."

Click "Enable."

It should show something like this now.

So, the API is now enabled. The only thing needed now is the API key. Click "Credentials," then "Create Credentials," and then "API key."

Copy and paste the generated API key somewhere safe, then log off.
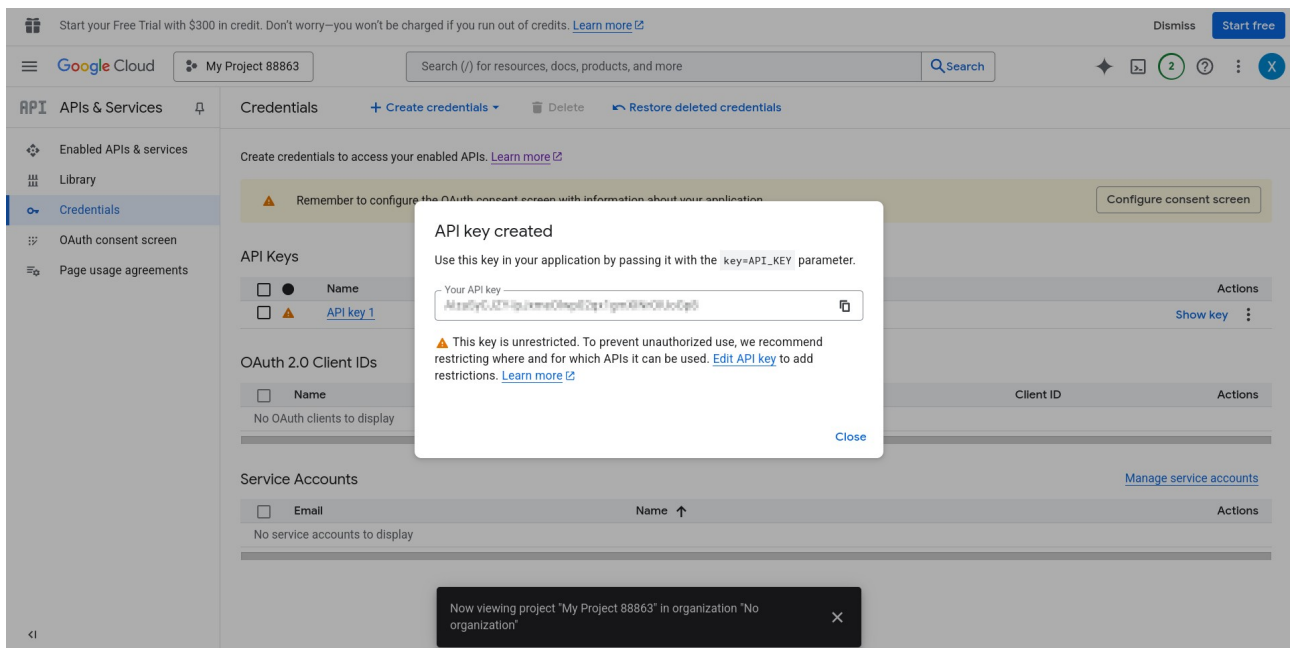
Next, open the `index.html` file, look for the `apiKey` variable (around line 63 or so), paste the API key into it, then save and close.
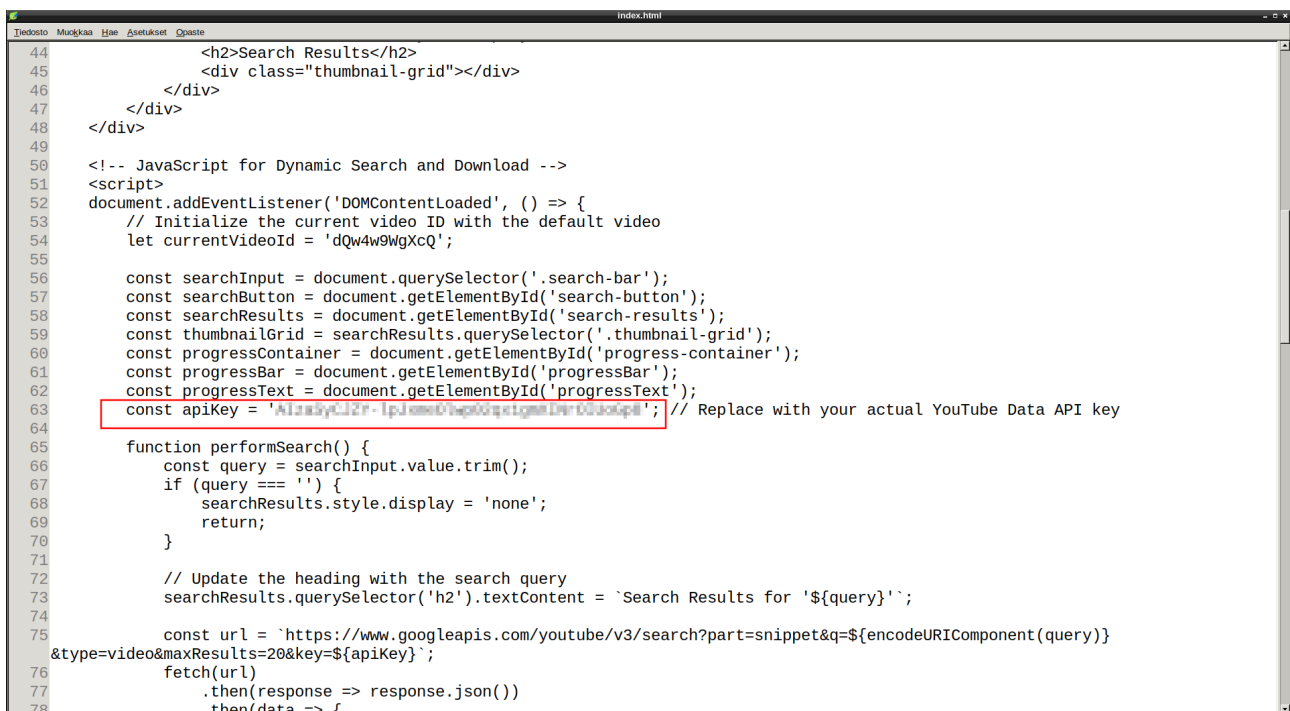
Start the web server...

```
/tmp/LocalTube $ python -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Then open http://localhost:8000 in your browser.If it works, you should see something like this. You can let Ricky boy sing or search for, let's say, "predator" in the search bar and then click "Go to Results."
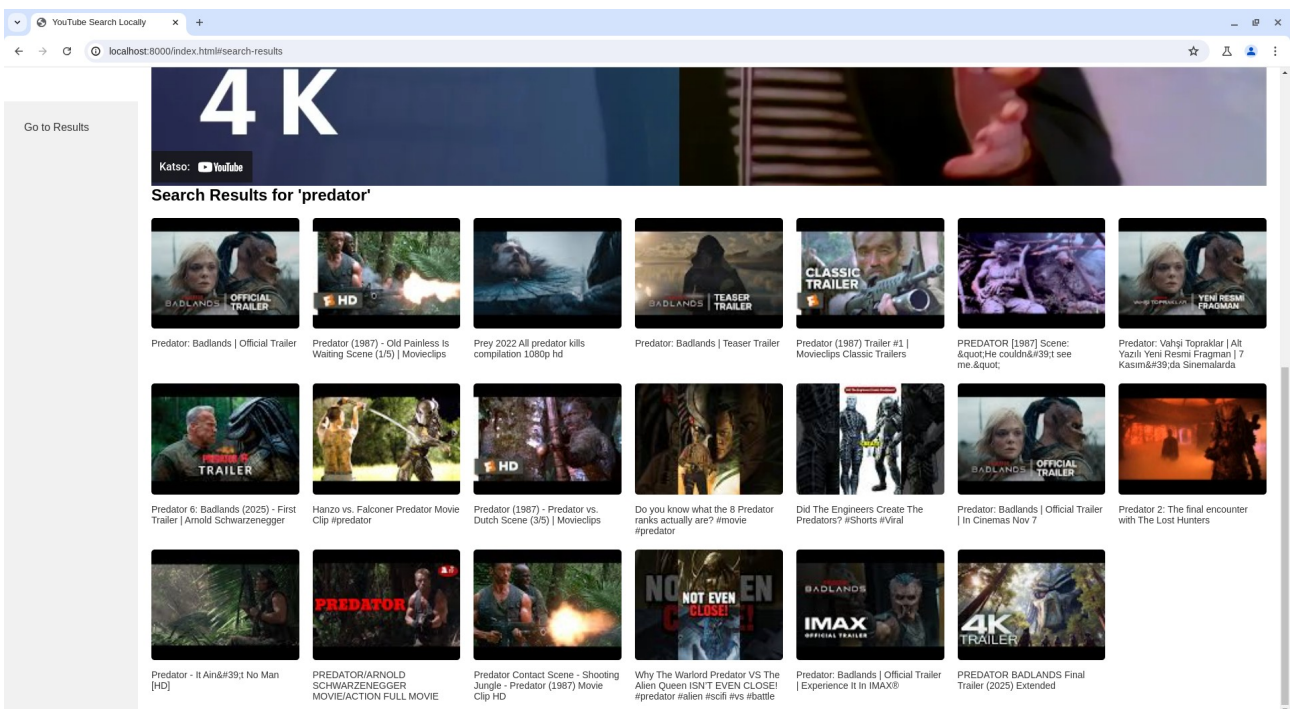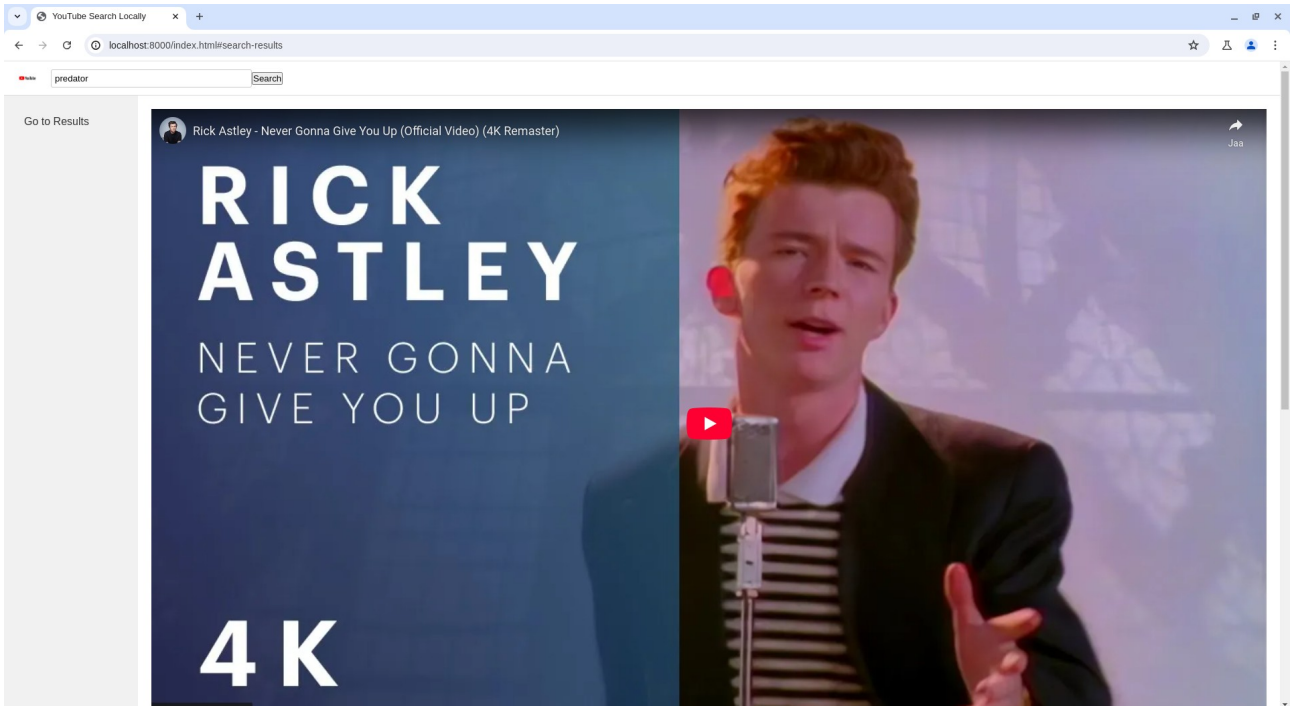
Congrats, you now have a local YouTube front-end without Google Analytics or other user-tracking nonsense. Of course, the actual search results and video stream still come from YouTube. If you're adventurous enough, you could even put this on a remote VPS server. The catch? Your free API key allows you to make about 100 requests per day, but it should be more than enough for personal use. BTW, the search results are capped to 20 (hey! this is quick & dirty remember?) but Im sure you can "hack" it to give more if you want.

Enabling Optional Video Downloading in LocalTube

This optional feature lets you download videos (e.g., in 1080p or higher) through LocalTube, in addition to streaming. Follow these steps to set it up:

1. Prerequisites
   - Ensure Python 3.x is installed (check with: `python3 --version`).
   - Ensure pip (Python package installer) is available.
2. Create a Virtual Environment
   - Open a terminal in the `LocalTube` directory and run:
     ```bash
     python3 -m venv venv
     ```
3. Activate the Virtual Environment
   - Activate it by running:
     ```bash
     source venv/bin/activate
     ```
   - Your terminal prompt will change (e.g., `(venv)` appears).
4. Install Required Packages
   - Install the necessary packages using the provided `requirements.txt` file:
     ```bash
     pip install -r requirements.txt
     ```
   - This installs `flask` and `yt-dlp`.
5. Run the Flask Application
   - Start the Flask server with:
     ```bash
     python3 app.py
     ```
   - The server typically runs on `http://localhost:8000`.
6. Access LocalTube with Downloading
   - Open your browser and go to `http://localhost:8000` (or the address shown in the Flask output).
   - You can now stream and download videos.
7. Deactivate the Environment
   - When finished, deactivate the virtual environment by running:
     ```bash
     deactivate
     ```

NoteThis setup is optional and only required if you want to download videos. For basic streaming, you can simply use `python3 -m http.server 8000` without these steps.

Go to Results

Katso: YouTube

Download

: 31.3% (Speed: 2.86 MiB/s, ETA: 01:15)

Search Results for 'Mad Max'

Mad Max: Fury Road - Official Main Trailer [HD]

Mad Max: Fury Road (2015) - The chase begins (1/10) (slightly edited) [4K]

Mad Max - The Nightrider [HD]

⊘ Mad Max 2's Most BRUTAL (and Hilarious) Fail Ever! #WastelandChaos"

MAD MAX Full Movie 2025: Furiosa Road | Superhero FXL Fantasy Movies 2025 in English (Game Movie)

Official Trailer: Mad Max (1979)

FURIOSA : A MAD MAX SAGA | OFFICIAL TRAILER #1