

Shorts Studio Setup Guide

AI Consultant

SYSTEMS · 18 MIN READ · AUG 10, 2026

This is for anyone posting Shorts who's tired of scripting, recording, editing, and uploading by hand. Shorts Studio turns a single topic into a finished, captioned YouTube Short, script, AI voice, AI generated scenes, and a live upload, all from one local app. Setup is one time, after that every new Short is just type a topic, hit Generate, then Publish. Start with Step 1 below and work through Setup until all four connections show Ready.

WHAT YOU'LL SET UP

By the end of this guide, Shorts Studio will take a single topic and turn it into a finished, captioned YouTube Short, script, narration, AI generated scenes, and a published upload, without touching CapCut or uploading anything by hand.

Along the way you'll install Node.js and ffmpeg, connect ElevenLabs for narration, connect Kling for AI images and video, and connect your YouTube channel through Google Cloud. All of it is one time setup. After that, every Short is just type a topic, hit Generate, then Publish.

STEP 1: INSTALL NODE.JS

Node.js is the engine that runs Shorts Studio on your computer. Install the LTS version, that's the stable release meant for regular use, not the experimental one.

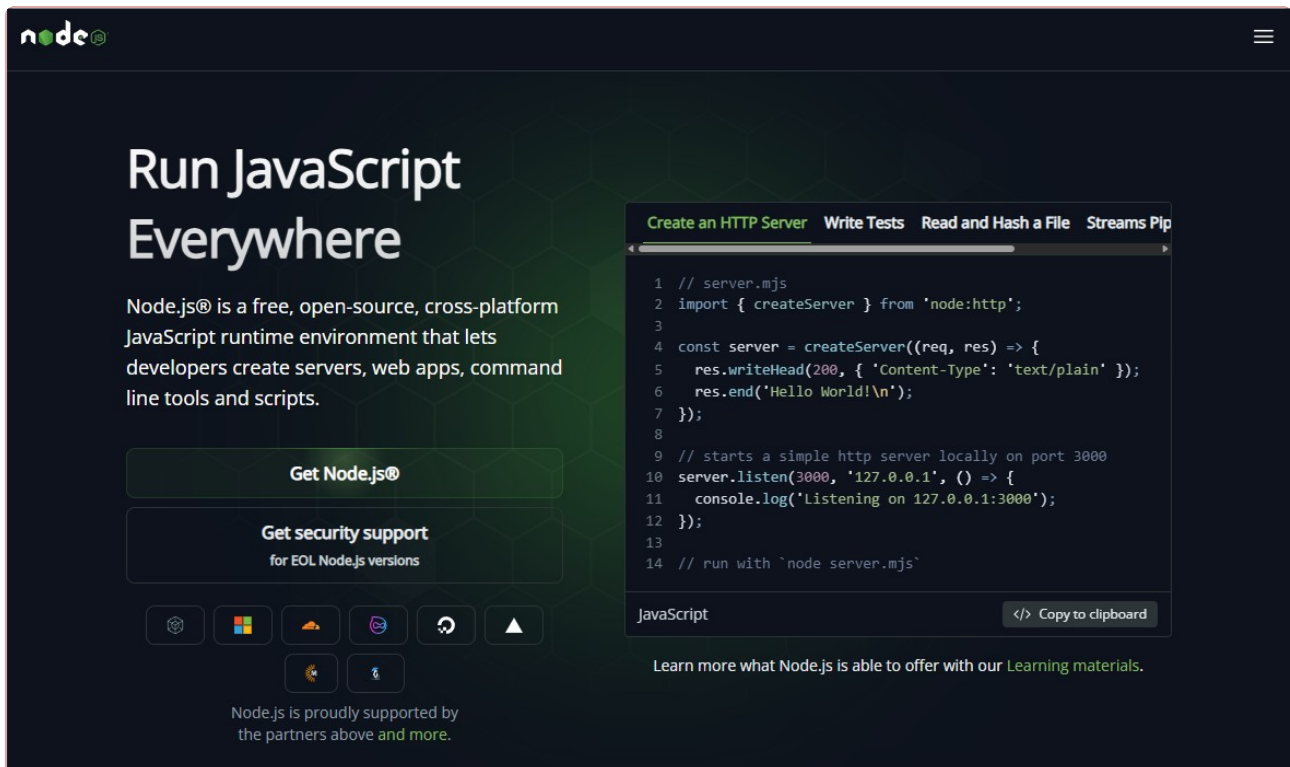
Download it from nodejs.org.

Open a new terminal and verify it installed correctly:

TERMINAL

```
node -v  
npm -v
```

Both commands should print a version number. If either one says command not found, close and reopen your terminal, then try again.



STEP 2: INSTALL FFMPEG

ffmpeg is what actually stitches your narration and AI generated scenes into one finished video file. Without it, Shorts Studio can build everything else but can't assemble the final MP4.

Windows:

CODE

```
winget install ffmpeg
```

Mac:

TERMINAL

```
brew install ffmpeg
```

Linux:

TERMINAL

```
sudo apt install ffmpeg
```

Open a new terminal after installing, then verify:

CODE

```
ffmpeg -version
```

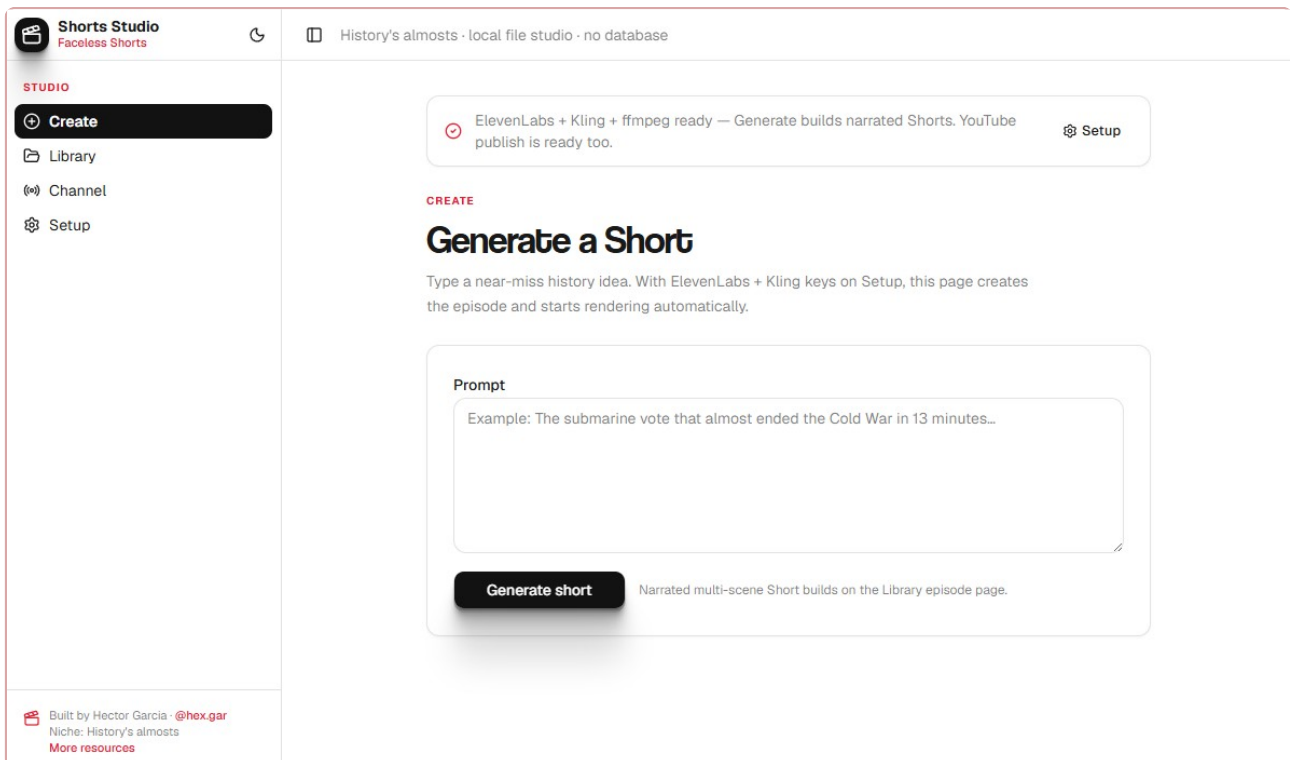
STEP 3: START SHORTS STUDIO

In the Shorts Studio project folder, install dependencies once, then start the app:

TERMINAL

```
npm install  
npm run dev
```

Leave that terminal window open and running, Shorts Studio needs it active the whole time you're using the app. Open localhost:3000 in your browser, that's the Shorts Studio interface.



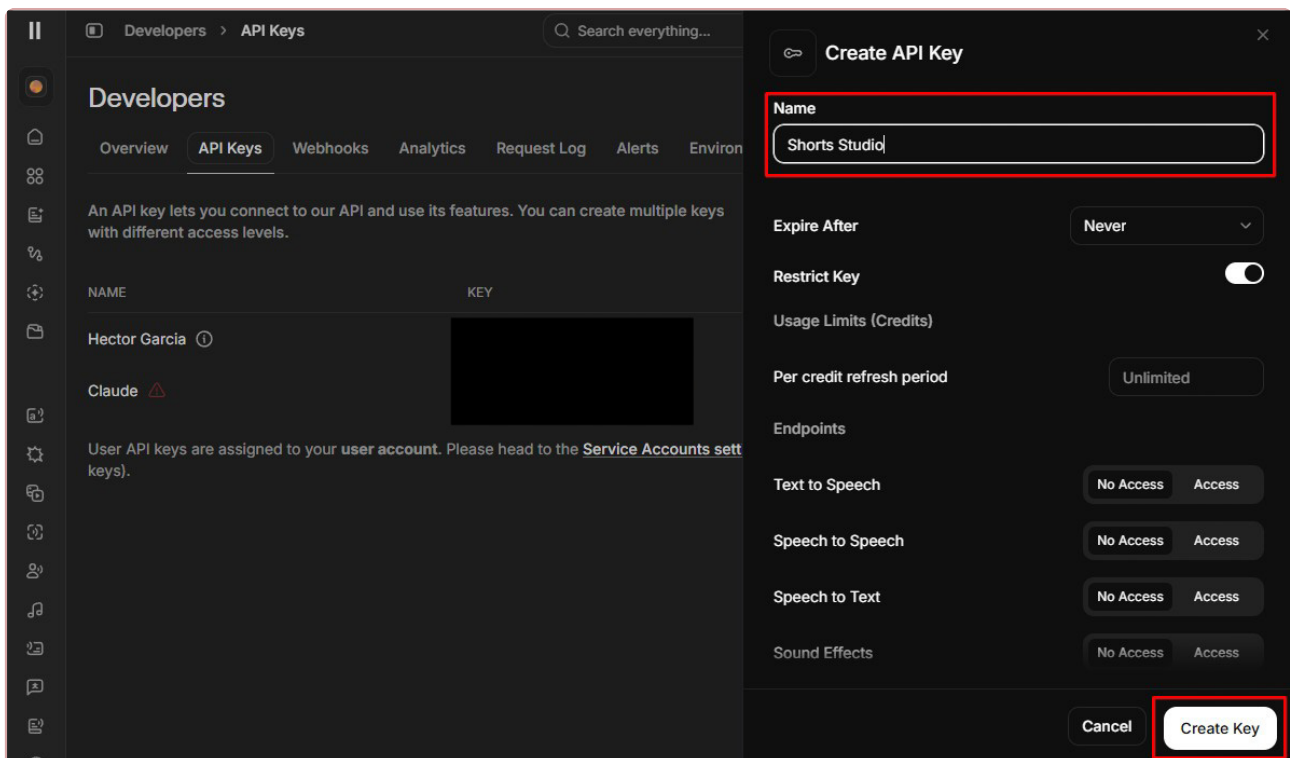
STEP 4: CONNECT ELEVENLABS (VOICE)

This is the fastest connection, about 2 minutes. ElevenLabs generates the narration voice for every Short.

Sign up or log in at [ElevenLabs](#), then create an API key: open Developers, then Profile and API key, then Create API key. Copy it somewhere safe.

Paste it into Shorts Studio's Setup panel, under ElevenLabs, then click Save.

Voice ID is optional, the default voice works fine to start.



STEP 5: CONNECT KLING (IMAGES AND VIDEO)

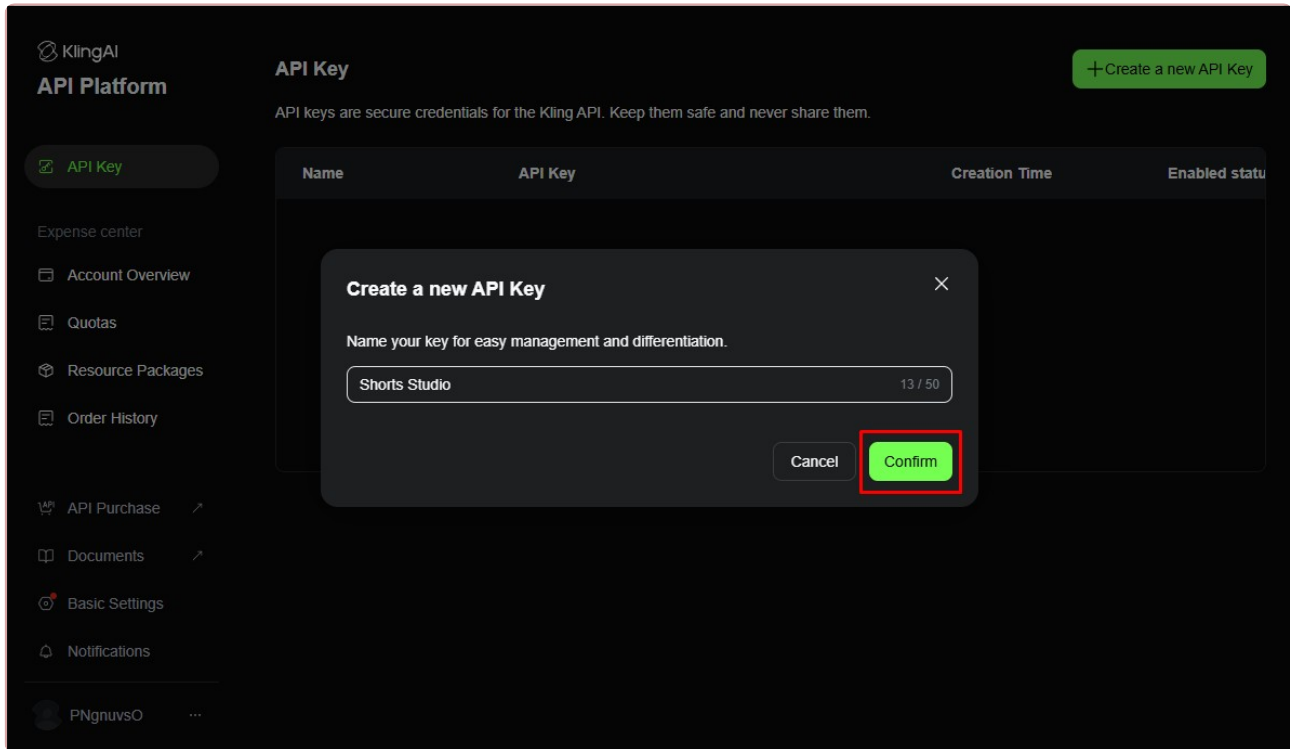
This one takes longer, about 10 minutes, and it's the step most people get wrong: a normal Kling membership does not pay for API access. The API runs on a separate credit system called developer packs, so even if you're already subscribed to Kling, you still need to buy these separately.

Create an API key at [Kling dev API key](#) and copy it, you'll only see it once.

At [Kling dev pricing](#), buy a small Image pack and a small Video pack, you need both, Shorts Studio uses Kling for still images and for the short video clips.

Paste the key into Setup, under Kling, then Save.

Important: If Generate says "Not enough Kling API units," don't check Order History, check Expense Center, then Quotas / Resource Packages instead, that's where your actual remaining balance shows.

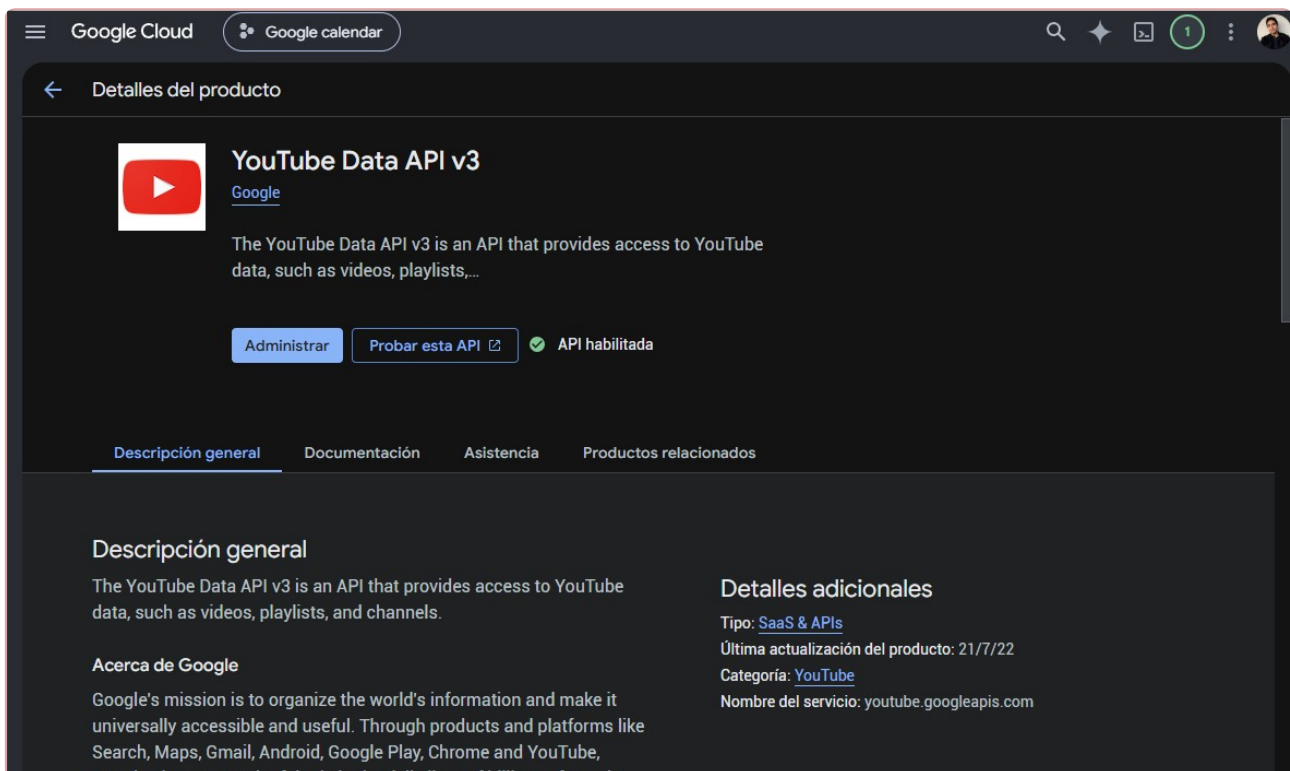


STEP 6: CREATE A GOOGLE CLOUD PROJECT

Connecting YouTube is the longest step, about 15 minutes, but it's one time only. After this, publishing a Short is a single button click.

Open [Google Cloud Console](#) and create a new project, or pick an existing one, the name doesn't matter, Shorts Studio works fine.

Enable the [YouTube Data API v3](#) for that project, this is what lets Shorts Studio upload videos to your channel on your behalf.



STEP 7: CREATE AN OAUTH CLIENT

The next tool you'll use, Google's OAuth Playground, specifically needs a Web application client, not a Desktop client and not a plain API key. Using the wrong type here is the most common setup mistake.

In Google Cloud Console, go to APIs & Services, then Credentials. If it asks you to configure a consent screen first, choose External, that's fine for your own channel, and add yourself as a Test user while it's in Testing mode.

Click Create credentials, then OAuth client ID. For Application type, choose Web application.

Under Authorized redirect URIs, add this exactly:

CODE

```
https://developers.google.com/oauthplayground
```

Click Create, then copy the Client ID and Client Secret, you'll need both in the next step.

Note: If you already created a Desktop client by accident, either edit it to add the redirect URI above if Google allows it, or just create a new Web application client instead.

STEP 8: GET YOUR REFRESH TOKEN

Open the [Google OAuth Playground](#).

Click the gear icon in the top right, then enable "Use your own OAuth credentials." Paste in the Web application Client ID and Client Secret you just copied, then close the panel.

Warning: Don't skip that step. If you leave it off, Google uses its own Playground app instead of yours, and you'll get errors later that trace back to this exact spot.

In the API list on the left, find YouTube Data API v3 and select `https://www.googleapis.com/auth/youtube.upload`.

Click Authorize APIs, sign in with the Google account that owns your YouTube channel, then Allow.

Click Exchange authorization code for tokens, then copy the Refresh token that appears.

Note: If you see "Error 400: redirecturimismatch," your Web client is missing the oauthplayground redirect URI from Step 7, add it, wait about a minute for it to take effect, then try again.

STEP 9: FINISH YOUTUBE SETUP

Back in Shorts Studio's Setup panel, under YouTube, paste in your Client ID, Client Secret, and Refresh Token, then Save. YouTube should switch to Ready.

Ready is Shorts Studio's way of confirming a connection actually works. Once all four items, ElevenLabs, Kling, ffmpeg, and YouTube, show Ready, you're fully set up and Generate and Publish will work end to end.

Shorts Studio

Faceless Shorts

History's almos... · local file studio · no database

STUDIO

Create

Library

Channel

Setup

21m00Tcm4TlvDq8ikWAM

Save ElevenLabs settings

KLING OPEN PLATFORM

Create API key

Direct Kling API (not a reseller). Create one API key at kling.ai/dev/api-key, buy a small **API unit pack**. Generate uses Kling image → Kling 2.1 image-to-video (5s, std).

API Key

Saved — paste to replace

Save Kling API key

YOUTUBE (OPTIONAL)

Enable YouTube Data API

Client ID

Client Secret

Refresh Token

Save YouTube keys

Built by Hector Garcia · @hex.gar

Niche: History's almos...

More resources

MAKE AND PUBLISH YOUR FIRST SHORT

- 1 Go to Create and type a topic, something like "The telegram that almost started WW3"
- 2 Click Generate short, then stay on the episode page until the status changes to Ready
- 3 While it runs, Shorts Studio writes the script, narrates it, builds five AI scenes, and stitches everything into a captioned vertical video. This can take several minutes, keep the tab open and keep `npm run dev` running
- 4 Preview it with the phone player on the left. If it's black, hard refresh (`Ctrl+Shift+R`) and press play, or just use Open/download video instead
- 5 Click Publish to YouTube and wait for the upload to finish
- 6 Open YouTube Studio, your Short is there, set to unlisted by default, review it, then switch it to Public when you're ready

Shorts Studio

Faceless Shorts

History's almos... · local file studio · no database

← Library

Published

Delete

STUDIO

Create

Library

Channel

Setup

EPISODE

The telegram that almost started WW3

The telegram that almost started WW3



GENERATE

Builds a ~45-60s narrated Short: story beats → ElevenLabs VO → 5 Kling scenes (image → 2.1 12V) → ffmpeg mux with captions.

Hook

Setup

The almost

Twist

Button

Done

Done

Done

Done

Done

Regenerate

SCRIPT

Template "History's almos..." narration (no LLM key).

HOOK

What if one wrong move almost rewrote the world? This is the story of The telegram that almost started WW3.

SETUP

The world looked stable. People went about their day. Nobody realized how close history

Built by Hector Garcia - @hex.gar

Niche: History's almos...

More resources

EVERYDAY CHECKLIST

Once Setup is fully Ready, this is the entire loop for every new Short.

- ☐ Start `npm run dev` if Shorts Studio isn't already running
- ☐ Go to Create, type a topic, click Generate short
- ☐ Wait for the status to hit Ready
- ☐ Click Publish to YouTube
- ☐ In YouTube Studio, switch it from unlisted to public when you're ready

GOOD TO KNOW

You only need to top up Kling when your units actually run out, there's no ongoing subscription cost beyond that.

Want the same Short on TikTok or Reels too? Download the same MP4 from Shorts Studio and upload it directly, no extra generation needed.

COST PER SHORT

SERVICE	WHAT YOU USE
Kling	About 5 images plus five 5 second video clips, from your Image and Video packs
ElevenLabs	About 45 to 60 seconds of narration
YouTube	Free, uploading to your own channel doesn't cost anything

TROUBLESHOOTING

PROBLEM	WHAT TO TRY
Publish button disabled	Finish the YouTube section in Setup, all three fields need to be filled in
YouTube upload failed	Generate a new refresh token in OAuth Playground, confirm you're added as a consent screen test user, and confirm the YouTube Data API is enabled
Not enough Kling API units	Buy an Image pack and a Video pack, then check Quotas under Expense Center
ffmpeg missing	Reinstall ffmpeg, check it's on your PATH, open a new terminal, then restart <code>npm run dev</code>
Player is black but status says Ready	Hard refresh and press play, or just download the MP4 directly
Generate seems stuck	Keep the episode page open, check your Kling units, and check the terminal running <code>npm run dev</code> for errors

USEFUL LINKS

ElevenLabs: [ElevenLabs](#)

Kling API key: [Kling dev API key](#)

Kling packs: [Kling dev pricing](#)

YouTube Data API: [Enable YouTube Data API v3](#)

OAuth Playground: [Google OAuth Playground](#)

ffmpeg: [ffmpeg downloads](#)

KEEP YOUR KEYS PRIVATE

Warning: Your keys stay on your own computer, in `data/secrets.json` or `.env.local`. Never share them or commit them to git.

WORK WITH ME

Business owner looking to automate?

Book a free 30-minute discovery call. We'll pressure-test fit, scope, and what to build first.

hexgarcia.com/free-call

Questions? Reply to the newsletter or reach out via email.

Hector Garcia

@hex.gar · contact@hexgarcia.com