

Open Montage Setup Guide

Install the free AI video pipeline, run your first render with zero API keys, and fix the common errors.

What this is and why the manual way drags

Open Montage is a free, open-source video pipeline that runs inside your AI coding assistant. You describe a video, it researches, scripts, gathers footage, voices it, and renders a finished cut. This guide gets it installed and running your first video without paying for a single API key.

The old way, I would open a doc, write a script, hunt for stock clips, record a voiceover, then drop it all into an editor and cut for hours. Four separate tools, four context switches, and most of the day gone before anything rendered.

Open Montage collapses that into one prompt. It runs on Remotion and FFmpeg, ships with 12+ production pipelines, and lets you keep an approval gate at each stage so you stay in control. It still needs some tuning on the first few runs, so treat this as calibration, not a hands-off machine.

01

How it actually works

1. Research and script. The agent searches YouTube, Reddit, news, and academic sources, then writes a script grounded in what it found instead of guessing.
2. Assets. It generates or pulls the images, video clips, narration, music, and sound effects your scene plan calls for, picking each provider by a scored ranking.
3. Edit and compose. It assembles the timeline, burns captions, mixes audio, and renders the final file through Remotion and FFmpeg, then self-reviews the output with ffprobe before calling it done.

02

The one thing almost everyone gets wrong

People assume they need a wall of paid API keys before it will produce anything. They do not.

WATCH OUT

Every API key in Open Montage is optional. Run ``make demo`` first with no keys at all. You get Piper for offline voiceover, free footage from Archive.org, NASA, and Wikimedia Commons, and free stock from Pexels, Pixabay, and Unsplash, all rendered through Remotion and FFmpeg. Add paid providers later, only for the pieces where free quality is not enough.

SOURCE CODE

The full repo, released under AGPLv3:

<https://github.com/calestio/OpenMontage>

PIPELINE ORDER

research -> proposal -> script -> scene_plan -> assets -> edit -> compose

Each stage is driven by a director skill (a markdown instruction file) under skills/pipelines/. Approval gates sit at proposal, script, scene_plan, assets, and publish.

12+ PIPELINES YOU CAN CALL

Animated Explainer
Animation & Motion Graphics
Avatar Spokesperson
Cinematic Trailers
Clip Factory (batch short-form extraction)
Documentary Montage (real footage)
Hybrid (footage + AI visuals)
Localization & Dubbing
Podcast Repurposing
Screen Demo
Talking Head
Character Animation

SITES SHOWN IN THE VIDEO

- OpenMontage on GitHub
<https://github.com/calestio/OpenMontage>

Set it up in Claude Code or Codex

1. Check your prerequisites first: Python 3.10 or newer, FFmpeg, and Node.js 18 or newer. On a Mac, ``brew install ffmpeg node python`` covers them.
2. Clone the repo into wherever you keep projects: ``git clone https://github.com/calestio/OpenMontage.git`` then ``cd OpenMontage``.
3. Run the one-shot installer: ``make setup``. If you do not have make, run the manual block below instead.
4. Open the OpenMontage folder in your AI coding assistant. It reads CLAUDE.md for Claude Code or CODEX.md for Codex automatically, both of which point at the shared AGENT_GUIDE.md, so the agent already knows the contract.
5. API keys are optional. When you do want a paid provider, ask your assistant to add the key to your shell profile (for example ELEVENLABS_API_KEY or PEXELS_API_KEY), then open a new terminal so it loads. The repo also reads them from the .env file it created from .env.example.
6. Do a zero-key smoke test: ``make demo`` renders demo videos with no keys, which proves FFmpeg, Node, and Remotion are all wired up.
7. Then point it at a real prompt inside your assistant, for example: Make a 60-second animated explainer about how neural networks learn. A good first run pauses at the proposal gate, shows you 2-3 variants with cost estimates, and only renders after you approve.

04

The prompts I actually run

ZERO-KEY EXPLAINER

Make a [LENGTH]-second animated explainer about [TOPIC]. Use free providers only.

REAL-FOOTAGE DOCUMENTARY

Make a [LENGTH]-second documentary montage about [SUBJECT]. Use real footage only, no narration.

REFERENCE-DRIVEN

Here is a YouTube video: [URL]. Analyze its transcript, pacing, and style, then propose 2-3 differentiated variants with cost estimates before you produce anything.

REPURPOSE A LONG VIDEO

Take [PODCAST OR LONG VIDEO FILE] and use the clip factory to extract [NUMBER] short-form clips, each under 60 seconds, captioned.

When it breaks

ERROR	FIX
make: command not found	Use the manual install block: create the venv, pip install -r requirements.txt, npm install inside remotion-composer, pip install piper-tts, then cp .env.example .env.
ffmpeg or ffprobe not found	Install FFmpeg (brew install ffmpeg on macOS) and reopen the terminal. Open Montage uses ffprobe in its post-render self-review, so it needs both.
npm install fails in remotion-composer	Confirm Node.js is 18 or newer with node -v, then delete node_modules and package-lock.json and run npm install again.
Output looks like a slideshow	That is the slideshow-risk score flagging static footage. Ask for real motion footage or an AI video provider, or switch to the Documentary Montage or Hybrid pipeline.
Provider or API key error	Keys are optional. Remove the failing provider or ask the agent to fall back to a free one (Piper, Pexels, Archive.org). Add the key to your shell profile and open a new terminal if you do want it.
Run stops at an approval gate	That is by design. Open the Backlot board (python -m backlot open) and approve the proposal, script, scene plan, or assets to continue.
Cost estimate is higher than expected	Set a spending cap in .env and lean on free providers. The default per-action approval threshold is \$0.50, so anything pricier asks first.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>