

SETUP GUIDE

Capture Every Content Idea Automatically with Telegram

Build a Telegram bot that catches your ideas, sorts them into your content system, and saves them to Airtable — while you are walking, driving, or standing in the shower.

01 — WHAT YOU'LL LEARN

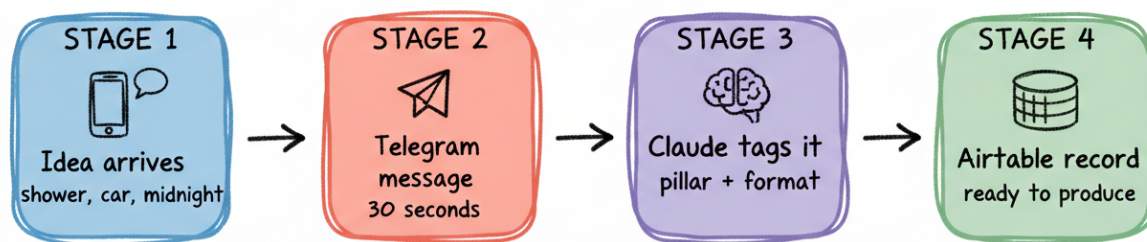
- ✓ Why a Telegram bot with Claude behind it beats every notes app for content idea capture
- ✓ How to build the bot, connect it to Airtable, and make it understand your content system
- ✓ How to keep it running forever on your Mac without ever thinking about it again

IN THIS GUIDE

- 1 Create Your Bot with BotFather** Five minutes. One conversation. You end up with a bot token and a working Telegram bot.
- 2 Get Your API Keys** Three credentials you need before writing a line of code. Get them now.
- 3 Write the Bot with Claude Code** You do not need to write this yourself. One prompt. Claude Code does the work.
- 4 Test It** Run it from your terminal and send it some ideas. If the records appear in Airtable, you are done.
- 5 Make It Run Forever** Use macOS launchd to keep the bot alive in the background, even after restarts.
- 6 Troubleshoot** Common errors, plain explanations, and exactly how to fix them.

02 — WHY THIS EXISTS

Idea to Database Flow in 30 Seconds



Not one idea lost in weeks

Idea to Database Flow in 30 Seconds

My brain generates ideas faster than I write them down. So I stopped trying to write them down.

I built a Telegram bot instead. One message — text, voice note, half a thought at 2am — and the bot picks it up, tags the content pillar, suggests a format, and parks it in Airtable before I have put the phone down. Ideas in the shower. Ideas on the school run. Ideas while burning dinner. All captured. All in the right place.

You already have a notes app. Here is why it does not work for content ideas: notes apps capture words. They do not organise them, connect them to your production workflow, or remember any context about what you are building. You end up with 200 untriaged notes and no system for turning any of them into posts. I know because I had exactly that.

A Telegram bot with Claude behind it does something different. It understands your content system — your pillars, your formats, your voice. When you type **"had an idea about the hardest part of posting every day"**, it does not save those words in a folder. It tags the pillar, suggests a format, and saves a structured record to Airtable. The notes app is a black hole. The bot is a collaborator.

QUICK TIP

Other creators solve this with Otter.ai, Notion AI, or Zapier email routing. They all work to a point. What they lack is the conversation. The bot helps you shape an idea before it goes into the database. That back-and-forth is the difference between capturing ideas and refining them.

03 — HOW IT WORKS

The whole thing is six steps, and the whole thing takes about 30 seconds from idea to saved record. You never open a laptop.

- 01 You open Telegram on your phone and message your bot
- 02 The bot sends your message to the Claude API, along with a system prompt that describes your content system
- 03 Claude reads the message, works out what type of content idea it is, and calls a tool to save it
- 04 The tool creates a record in Airtable with the idea text, pillar, format suggestion, and status
- 05 Claude sends you a confirmation back in Telegram: **"Saved. Parked in Inbox under KS-4."**

Because the bot keeps conversation history in memory, you also get proper back-and-forth. Send a half-formed thought, the bot asks one clarifying question, you answer, it saves the sharpened version. A conversation, not a form.

04 — STEP 1 — CREATE YOUR BOT WITH BOTFATHER

BotFather is Telegram's official tool for creating bots. Every Telegram bot in existence started here. The whole thing takes five minutes.

INSTRUCTIONS

1. Download Telegram on your phone if you have not already (free, all platforms)
2. Search for **@BotFather** — look for the blue verified tick
3. Send: **/newbot**
4. Give your bot a display name (e.g. My Ideas Bot)
5. Give it a username ending in "bot" (e.g. @my_ideas_capture_bot)
6. BotFather sends you a bot token — a long string of characters starting with numbers and a colon
7. Copy it. Store it somewhere safe. This is your bot's password.

QUICK TIP

Set a description and a profile photo in BotFather using **/setdescription** and **/setuserpic**. Small thing. Makes the bot feel like a real tool rather than a test account, and you will use it more if it looks right.

IMPORTANT

Anyone who knows your bot's username can find it and send it messages. Your bot code must check the chat ID of every incoming message and ignore anything that does not come from you. You will set this up in Step 3.

05 — STEP 2 — GET YOUR API KEYS

You need four credentials before writing any code. Get them all now so you are not hunting for them mid-build.

Claude API Key — Go to console.anthropic.com, create an account or sign in, go to API Keys and create a new key. Store it as `ANTHROPIC_API_KEY`. Pay-as-you-go, no subscription. At roughly eight ideas a week, expect about £2-3 a month.

Airtable Token — Go to airtable.com/create/tokens. Create a token with scopes `data.records:read` and `data.records:write`. Select the base you want the bot to write to. Store it as `AIRTABLE_TOKEN`.

Airtable Base and Table IDs — Open your Airtable base. The URL contains `appXXXXXXXXXXXX` (your Base ID) and `tblXXXXXXXXXXXX` (your Table ID). Store them as `AIRTABLE_BASE_ID` and `AIRTABLE_TABLE_ID`.

Telegram Chat ID — Message your new bot anything — "hello" is fine. Then open this URL in a browser: <https://api.telegram.org/bot/getUpdates>. Find the "chat" object in the JSON response. The "id" field is your chat ID. Store it as `TELEGRAM_CHAT_ID`.

06 — STEP 3 — WRITE THE BOT WITH CLAUDE CODE

You do not need to be a developer. You do not need to write this code yourself. Open Claude Code and tell it what you want.

PROMPT FOR CLAUDE CODE

Build a Telegram bot in Python. Single file. It does the following:

1. Listens for messages from Telegram using polling (python-telegram-bot library)
2. Checks the chat ID — if it is not `MY_CHAT_ID`, ignore the message
3. Sends the message text to the Claude API with a system prompt describing my content system
4. My content system has three pillars [describe yours] and several format codes [list yours]
5. Claude has one tool: `save_idea`, with parameters: `title` (string), `core_idea` (string), `pillar` (string), `format_suggestion` (string), `status` (string, default "Inbox")
6. When Claude calls `save_idea`, the bot creates a record in Airtable via the REST API
7. The bot sends Claude's response back to the user via Telegram
8. Conversation history is kept in memory so multi-turn exchanges work

Load API keys from a `.env` file using `python-dotenv`. Variables: `TELEGRAM_BOT_TOKEN`, `ANTHROPIC_API_KEY`, `AIRTABLE_TOKEN`, `AIRTABLE_BASE_ID`, `AIRTABLE_TABLE_ID`, `TELEGRAM_CHAT_ID`.

Claude Code will generate the full script. Review it, run it, and adjust the system prompt until the categorisation fits your content system. The system prompt is the thing that matters most — it is what turns a dumb message relay into something that understands what you are building. It should contain your pillars, your format codes, and instructions to use your exact words rather than rewrite them.

IMPORTANT

Airtable field names in your code must match your base exactly — capitalisation, spaces, everything. A 422 error from Airtable almost always means a field name is wrong. Check it against your actual base before you spend an hour debugging something else.

07 — STEPS 4 AND 5 — TEST IT, THEN MAKE IT PERMANENT

Run the bot from your terminal to test it:

REFERENCE

```
PYTHONUNBUFFERED=1 python3 your_bot_script.py
```

Message your bot from Telegram. If a record appears in Airtable, the bot works. Now you need to make it run permanently in the background, restart itself if it crashes, and start automatically when you log in. macOS has a built-in tool for this called launchd. No extra software needed.

INSTRUCTIONS

1. Create a wrapper shell script (start-bot.sh) that loads your .env file and starts the bot. Ask Claude Code to write this for you — tell it the path to your project folder and your script name.
2. Make it executable: **chmod +x start-bot.sh**
3. Ask Claude Code to generate a launchd plist file. Tell it: your label (e.g. com.yourname.telegram-bot), the full path to start-bot.sh, RunAtLoad true, KeepAlive true, and log paths at /tmp/telegram-bot.log
4. Save the plist to ~/Library/LaunchAgents/
5. Load it: **launchctl bootstrap gui\$(id -u) ~/Library/LaunchAgents/com.yourname.telegram-bot.plist**

Check if it is running — `launchctl list | grep telegram` — exit status 0 means running

Stop it — `launchctl bootout gui$(id -u)/com.yourname.telegram-bot`

Check the log — `cat /tmp/telegram-bot.log` — this is where all errors will appear

08 — STEP 6 — TROUBLESHOOTING

Things will go wrong. Here is what they mean.

Bot does not respond at all — Check TELEGRAM_BOT_TOKEN is correct and the bot is not already running somewhere else. Telegram only allows one active polling connection per bot. If there are two copies running, messages go to whichever one answers first — or neither. Kill all copies with: `kill -f your_bot_script.py`

401 Unauthorized on Claude API — Your ANTHROPIC_API_KEY is wrong, expired, or not loading from the .env file.

401 on Airtable — Your AIRTABLE_TOKEN is wrong or does not have write permissions on that base.

422 on Airtable — A field name in your code does not match your Airtable base. Field names are case-sensitive and space-sensitive. Open the base, copy the exact names, and update the code.

Operation not permitted with launchd — macOS is blocking the script. Fix it in System Settings — Privacy and Security — Full Disk Access — add Terminal. You only do this once.

.env not loading — The bot starts but every API call fails. Check the cd command in your wrapper script lands in the folder containing .env, the file exists, and there are no trailing spaces or quotes around values.

QUICK TIP

Voice notes are the mobile power move. Telegram lets you hold the mic button, speak, release — all while walking or cooking. Add voice note handling to your bot by asking Claude Code to include a transcription step using OpenAI Whisper. About 20 extra lines of code, and the whole system becomes fully hands-free. Cloud transcription costs roughly £0.005 per minute — so almost nothing.

09 — WHAT IT COSTS

Item	Monthly Cost
Telegram	Free
Claude API (approx. 8 ideas/week)	~£2-3
Airtable (free tier, up to 1,000 records)	Free
Running on your Mac	Free
Whisper transcription (optional, cloud)	~£0.50
Total	£2-4 per month

Less than a coffee a week. If you want 24/7 uptime even when your laptop is closed, Railway (railway.app) runs it in the cloud for roughly £5 a month extra. For most people running this on a Mac that is on during the day, that is an unnecessary expense.

When the system is running properly, here is what it changes: you stop building your content queue at your desk and start building it everywhere else. On walks, on the school run, in the supermarket at 11pm (the ideas there are terrible, but they are captured). When you sit down to produce content, the queue is already there. You did not build it that morning. You built it over weeks of messages, one idea at a time.

QUICK TIP

PS. The bot is not the clever part. The clever part is the system prompt — the bit that tells Claude your pillars, your formats, and what you are building. Get that right and the bot stops being a notepad. It starts being a collaborator. Spend time on the system prompt. Everything else is plumbing.

Want this for your team?

I run AI training and consulting at [Fruitful AI](#). [Book a thirty-minute call](#) on where the time goes in your team, and whether training is the thing that fixes it. Here are some of the tools I use to build my apps and solutions.

[Claude Code](#)
The AI coding agent I build with. I built my dashboard with it.

- BUILD**
- [GitHub](#)
 - [Vercel](#)
 - [Supabase](#)

- DATA AND AUTOMATION**
- [Airtable](#)
 - [n8n](#)
 - [Telegram](#)

- VIDEO AND SOCIAL**
- [HyperFrames](#)
 - [Wavespeed](#) *
 - [upload-post](#)
 - [ManyChat](#)

* affiliate link