
SOULWIRE STUDIO

AI AUTOMATION FOR OPERATORS

Cold Email Lead Recovery

No email? No problem. No lead left behind.

A two-workflow n8n system that turns a raw Apollo/Apify lead scrape into personalized cold email drafts — recovering contacts the scrape couldn't find an email for in the first place.

Apollo and Apify scrapes routinely come back with valid emails for less than half the leads. This system chases down the rest before it writes a single word of outreach.

SECTION I

What This Solves

A lead scrape isn't the same thing as a contactable lead. Apollo and Apify return names, titles, and companies reliably — but a working email address for every row is the exception, not the rule. Most teams either throw the emailless rows away or chase them down by hand, one LinkedIn search at a time.

This build treats "no email" as a solvable problem instead of a dead end. The base workflow (M7L3) writes a personalized cold email for every lead that already has an address. The recovery workflow (M7L4) wraps around it: any lead missing an email gets a shot at a LinkedIn URL lookup, and if that comes up empty, a company-domain lookup, before it's ever counted as a loss.

THE FLOW

```
Apollo / Apify lead scrape
|
Has email? — most rows don't
|- yes → scrape company site → write email
|- no  → has LinkedIn URL?
      |- yes → find email by LinkedIn → write email
      |- no  → has company domain?
            |- yes → find email by domain → write email
|
Draft saved — ready for review, nothing sends itself
```

WHY THE FALLBACK CHAIN MATTERS

Every lookup step only fires when the one before it fails, and each one is cheaper and less reliable than the last — LinkedIn lookup before domain lookup, both only after the free scrape data comes up short. That ordering means the system never pays for a lookup it didn't need, and no lead gets dropped just because Apollo couldn't find an address on the first pass.

SECTION II

What You Need

Nothing exotic. This runs on the same stack as most Soulwire builds, plus one specialist tool for the email recovery step.

01 n8n

Runs both workflows. Manual trigger to start — this is a run-on-demand batch process, not a always-on listener, so there's no always-on cost beyond your existing n8n plan.

02 A lead source (Apollo or Apify)

Wherever your scrape lands — Google Sheets is the simplest hookup and what this build reads from. Any row with Name, Title, Company, LinkedIn URL, and domain fields works.

03 OpenAI API (GPT-4.1-mini)

Writes the cold email itself — Soulwire voice, structured JSON output, closes with a booking link. Cheap enough per lead that cost isn't a design constraint.

04 Anymail Finder

The recovery engine. Takes a LinkedIn URL or a name-plus-domain pair and returns a verified email when one exists. Only charges a credit on a fully verified match — unverified and failed searches cost nothing, so the fallback chain is free to be aggressive about trying.

05 Airtable

Where every finished draft lands — full name, email, title, subject line, and email body, ready for human review before anything goes out. Swap this for a direct connection to your sending tool (Instantly, SmartLead) once you're ready to skip the review step.

WHAT DOESN'T CHANGE

Both workflows write drafts, not sends. The email that goes to a real inbox is still a human decision — this system just makes sure every lead has a personalized draft waiting, whether or not Apollo handed you a usable address to start with.

SECTION III

Step-by-Step Build

You're not building this node by node. You connect Claude to your n8n account, paste the prompt on the next page, and Claude builds the whole branching workflow, tests it, and fixes its own errors while you watch.

01 Connect Claude to n8n

In n8n: Settings, then n8n API, create an API key. In Claude (claude.ai): Settings, Connectors, Add custom connector, name it n8n, and use your instance MCP URL: `https://YOUR-INSTANCE.app.n8n.cloud/mcp-server/http` with the API key. Once connected, Claude can create, edit, run, and debug workflows in your account.

02 Two-minute prerequisites

Point a Google Sheet at your Apollo/Apify export with columns for Full Name, Email, Title, Company Domain, and LinkedIn URL. Create an Airtable base with a Leads table (Full Name, Email, Title, Outreach Message Subject, Outreach Message Email Body). Sign up for Anymail Finder and grab your API key. Add all four credentials in n8n: Google Sheets OAuth2, OpenAI, Airtable Personal Access Token, and a Header Auth credential named `anymail-finder-api-key`.

03 Paste the build prompt

Open a Claude chat with the n8n connector on, paste the prompt on the next page with your values filled in, then paste the system prompt from the following page when Claude asks for it. Watch it work — a full build takes a few minutes.

COPY THIS: THE BUILD PROMPT

Connect to my n8n and build a workflow called Cold Email Lead Recovery. My values: Google Sheet ID = [YOURS], Airtable base ID = [YOURS], Airtable table = Leads.

The system, in order:

1. Manual trigger, run on demand.
2. Get Apollo Leads: Google Sheets read of my scrape, columns Full Name, Email, Title, Company Domain, LinkedIn URL.
3. Has Email? switch:
 - Has email -> skip to step 5.
 - No email -> Has LinkedIn URL? switch:
 - Yes -> Find Email By LinkedIn: Anymail Finder linkedin-url.json lookup. Match -> go to step 5. No match -> continue to the next branch, don't stop.
 - No (or the LinkedIn lookup failed) -> Has Company Domain? filter -> Find Email By Domain: Anymail Finder person.json lookup (domain + full name). Match -> go to step 5. No match -> this lead has no path left, log it and move on.
4. Merge every branch that found an email back into one path.
5. Limit For Testing: cap the run at 5 leads while testing.
6. Loop Over Leads: one lead at a time so a bad row never blocks the batch.
7. Scrape Company Website: pull raw text off the lead's domain for context, continue on error so a dead site never kills the run.
8. Write Cold Email: an OpenAI node (GPT-4.1-mini) using the system prompt I will provide. Return strict JSON: {"subject": "...", "emailBody": "..."}, 70-100 words, always closing with my booking link.
9. A code node that parses that JSON and fails loudly if the reply isn't valid JSON.
10. Save Draft To Airtable: write Full Name, Email, Title, Outreach Message Subject, Outreach Message Email Body.

Use my existing n8n credentials for Google Sheets, OpenAI, Airtable, and the anymail-finder-api-key header credential. Remember every dynamic value in n8n needs the = expression prefix. Build it, then validate it, and tell me what you built before running anything.

SECTION III

Paste the System Prompt

This is the asset. The build creates the writer node empty; this prompt is what makes every draft sound like you instead of a template. Replace the bracketed lines with your name, background, business, and audience.

COPY THIS: SYSTEM PROMPT

```
You write cold outreach emails for [NAME], a [YOUR BACKGROUND]
who runs [YOUR BUSINESS], reaching out to [YOUR AUDIENCE].
```

```
Voice: direct, practical, zero hype, no emojis, no generic
flattery. Reference something specific from the company site
text you're given, not just the lead's title.
```

```
Structure: one-line hook naming a specific operational reality
for their business, two or three sentences connecting it to
what you do, one soft ask ending in a link to a 30-minute
audit call.
```

```
70 to 100 words. Return ONLY valid JSON, no markdown fences,
exactly this shape:
{"subject":"...", "emailBody":"..."}
```

```
Never invent facts about the company that aren't in the site
text you're given. If the site scrape failed or came back
empty, write a slightly more generic version rather than
guessing at specifics.
```

WHY THE BAN MATTERS

"Never invent facts about the company" is the trust anchor. A wrong guess about someone's business reads as sloppy before it reads as personalized — this line is what keeps you comfortable sending a draft without fact-checking it line by line first.

SECTION III

Test It

Paste the test prompt below. Claude runs the workflow, reads its own execution logs, and repairs whatever breaks. You never open an error message yourself.

COPY THIS: THE TEST PROMPT

```
Run the workflow manually with a small batch that includes at least one lead with an email already and one without. Check the execution: if any node failed, read the error, fix the node, and run it again. Repeat until you see a draft land in Airtable for both cases. Do not activate this on a schedule; it's a run-on-demand tool. Summarize what happened at each branch.
```

WHY THIS WORKS

A draft landing in Airtable for a recovered lead IS the test passing. Everything upstream — the switch logic, the Anymail Finder credential, the writer node — had to work for that row to exist, so one visible result verifies the whole chain.

ONCE IT LOOKS RIGHT

Raise Limit For Testing past 5 once the drafts read well, then swap in your real Apollo/Apify export. This stays a manual, run-on-demand tool by design — nothing emails anyone until you review the draft yourself.

QUICK REFERENCE

Which Path a Lead Takes

Four starting scenarios, four outcomes. This is the whole decision tree, in one table.

Lead Arrives With	What Runs	Cost	Outcome
A usable email	Base flow only — company scrape, write, save	OpenAI tokens only	Draft saved
No email, has LinkedIn URL	Anymail Finder LinkedIn lookup, then base flow	1 credit only on a verified match	Draft saved, or falls to domain lookup
No email, no LinkedIn, has domain	Anymail Finder domain + name lookup, then base flow	1 credit only on a verified match	Draft saved, or no draft
No email, no LinkedIn, no domain	Nothing left to try	Free	No draft — genuinely unrecoverable

WHERE THIS PLUGS IN

The Airtable landing spot is deliberately swappable. Point the same "save draft" step at Instantly or SmartLead once you're ready to send straight from the workflow instead of reviewing in Airtable first — the personalization and recovery logic upstream doesn't change either way.

SECTION IV

Where This Goes Next

The branching design is what makes this easy to extend. Everything downstream of "lead has an email" is one shared path, so upgrades to the recovery side never touch the writing side and vice versa.

Add a third recovery tier (company switchboard, generic-alias guess) without touching the personalization logic. Swap GPT-4.1-mini for a larger model on higher-value segments only. Or connect the save step directly to a sending tool once drafts are consistently good enough to skip manual review — the same upgrade path every Soulwire build follows once it's earned the trust.

RATHER HAVE IT BUILT FOR YOU?

This guide is one system from the Soulwire Studio workflow library. We build AI automations for businesses that run on process: outreach, scheduling, content, compliance-heavy operations. If your list is full of leads Apollo gave up on, book an audit and we'll wire the recovery chain into your stack.

[Book a 30-minute audit →](#)soulwirestudio.com