

// pillar 2 · part 1

Build the pipeline



n8n from scratch, and the tier that pulls live jobs straight from your target companies' career pages into a Google Sheet. Written for someone who has never opened n8n.

baswe.ai // become a software engineer

This is pillar two, the build, and it is the longest in the system, so it comes in two parts. Part one gets n8n running from a blank account and builds the tier that pulls live jobs straight from your target companies' career pages into a Google Sheet. Part two adds a market-wide keyword tier and a tier that flags jobs where you have a LinkedIn connection. This part is written for someone who has never opened n8n, and it ships with a companion code file, `Tier_A_Company_Career_Pages_n8n_code.js`, which you paste into two nodes.

What you're building

A small automation. Your list of target companies goes in, n8n hits the public job feed behind each company's careers page, and clean job rows come out into a sheet. There is no scraping and no logging in: you are calling the same feed the company's own careers page uses, which makes it the freshest possible source, and free.

Why this beats job boards: it is fresher than aggregators, it is free, it is filtered to your exact targets, and it is yours to control. The full pipeline has three tiers, and this part builds the first:

Tier A: company career pages. Pull jobs straight from your target companies (this part).

Tier B: market keyword search. Catch roles at companies not on your list (part two).

Tier C: connections match. Flag jobs where you already know someone (part two).

n8n in plain English

n8n is a visual automation tool. You drag "nodes" (steps) onto a canvas and connect them into a "workflow." Data flows left to right in small packets called "items." Each node takes items in, does one job, and passes items out. Here are the four node types you'll use:

Trigger. Starts the workflow when you click run.

Google Sheets. Reads your target list and writes the jobs back out.

HTTP Request. Fetches data from a web address, in our case each company's job feed.

Code. Runs a short script to reshape data. You paste our scripts in; you never have to write code yourself.

Get n8n running, from scratch

▪ Sign up

Go to n8n.io and start a free Cloud trial. Cloud means n8n runs for you in the browser, with nothing to install. You will land in your own workspace with a web address that is yours.

▪ Make a workflow

Create a new workflow. You will see a blank canvas with a plus button to add your first node. Nodes connect by dragging from the small circle on a node's right edge to the next node. Click any node to open it and see its settings, plus its input on the left and its output on the right.

▪ How you'll work

The rhythm is always the same: add a node, open it, set it, then click Execute (or Test step) to run just that node and see its output. You build the chain one node at a time, checking the output panel as you go. That is the whole game.

Set up your input sheet

The pipeline reads your target companies from a Google Sheet. Create a sheet, or use a Targets tab in your Pipeline tracker, with these columns: Company, ATS, Slug, and optionally Careers URL.

What ATS and Slug mean: almost every company hosts its jobs on an applicant tracking system (an ATS) like Greenhouse or Lever. The "slug" is the company's short name inside that ATS's web address. You read both straight off the company's careers-page URL:

SPOT THE ATS FROM A CAREERS URL

```
boards.greenhouse.io/acme    → ATS greenhouse, slug acme
jobs.lever.co/acme           → lever, acme
jobs.ashbyhq.com/acme        → ashby, acme
acme.recruitee.com            → recruitee, acme
apply.workable.com/acme      → workable, acme
careers.smartrecruiters.com/Acme → smartrecruiters, Acme
```

Fill in five to ten companies to start, and make at least one a Greenhouse company, because that is the simplest to get working first.

Connect n8n to Google Sheets

Add a Google Sheets node and create a Google Sheets credential. n8n walks you through signing in to Google and granting access, just once. After that, n8n can read and write your sheets. Point a "Get rows" operation at your Targets tab and click Execute: you should see your companies appear as items in the output panel. That confirms the connection works.

Build Tier A, node by node

Now the core. Five nodes in a chain. You paste the two Code node scripts from the companion file, and set everything else by clicking.

THE FIVE-NODE FLOW

Manual Trigger → Google Sheets (Targets) → Build Endpoint →
HTTP Request → Normalize → Google Sheets (Jobs)

▪ 1 · Trigger and read your targets

Add a Manual Trigger so you can run the flow on demand. Then add the Google Sheets "Get rows" node reading your Targets tab. Execute; each company shows up as one item.

▪ 2 · Build the endpoint (Code node)

Add a Code node, rename it exactly Build Endpoint, and set Mode to "Run Once for Each Item." Paste the first block from the companion file. It reads each company's ATS and slug and builds the exact public URL that returns that company's jobs. Execute; every item now carries an `endpoint_url`, like this:

```
endpoint_url: https://boards-api.greenhouse.io/v1/boards/acme/jobs?content=true
```

▪ 3 · Fetch the jobs (HTTP Request node)

Add an HTTP Request node. Set Method to GET. In the URL field, switch on expression mode and paste the expression below, so the node uses each item's own URL. Execute; you get back each company's raw jobs data.

```
{{ $json.endpoint_url }}
```

▪ 4 · Normalize (Code node)

Add a Code node named Normalize, with Mode "Run Once for All Items." Paste the second block. Each ATS returns a different shape, and this maps them all into one clean row shape (company, title, location, url, posted) and outputs one item per job. Execute; you get a tidy list of jobs.

▪ 5 · Write to your Jobs sheet

Add a Google Sheets node, operation "Append row(s)," pointed at a new Jobs tab. Map the columns to the normalized fields. Execute; your jobs land in the sheet.

You now have live jobs from your exact target companies flowing into a sheet, pulled straight from the source. That one tier already beats scrolling job boards.

Test first, then extend

Get the win first. Put one Greenhouse company in your Targets tab, run the whole chain top to bottom, and confirm jobs land in the sheet. Once that works, add more companies and more ATSs.

Troubleshooting is simple: click any node to see its output, which is how you find where something breaks. If the HTTP node returns nothing, the slug is almost always wrong, so recheck it against the careers URL. If a column comes out blank for a

non-Greenhouse source, that ATS names its fields a little differently; the companion file marks those spots with the word VERIFY, so do one live pull and adjust.

From our build: Greenhouse, Lever, and Ashby are confirmed working. Workable, Recrutee, SmartRecruiters, and Personio are mapped but worth a quick verify. Personio returns XML rather than JSON, so it needs one extra XML node before Normalize; skip it until your JSON sources are flowing.

What you have, and what's next

You now have a working Tier A: target companies in, live jobs out, into a Google Sheet, refreshable any time you hit Execute.

Part two adds Tier B, a market-wide keyword search so you also catch roles at companies not on your list, and Tier C, which matches jobs against your LinkedIn connections export to flag your warm leads. It also removes duplicates and puts the whole thing on a schedule so it runs on its own.

→ **Next: Pillar 2 · Part 2, the connections tier and going live.**