dapto

JEV SERIES / LINKEDIN TUTORIAL

Find potential customers in your LinkedIn network

Build LinkedIn Radar with Claude Code.
Use Jev to find people who fit your offer.

START WITH

Your connections and invitations

FINISH WITH

A filtered list of people to review and contact yourself

dapto

THE VALUE

Who is worth contacting first?

Say you are launching a service for operations leaders.

Example buyer: COO at an industrial company.

Chief Operating Officer

Industrial company — **Possible match**

Recruiter

Staffing agency — **Excluded by your rules**

No job title

Company listed — **Needs review**

Jev reads the available text. Your rules turn its answers into a shortlist you can inspect.

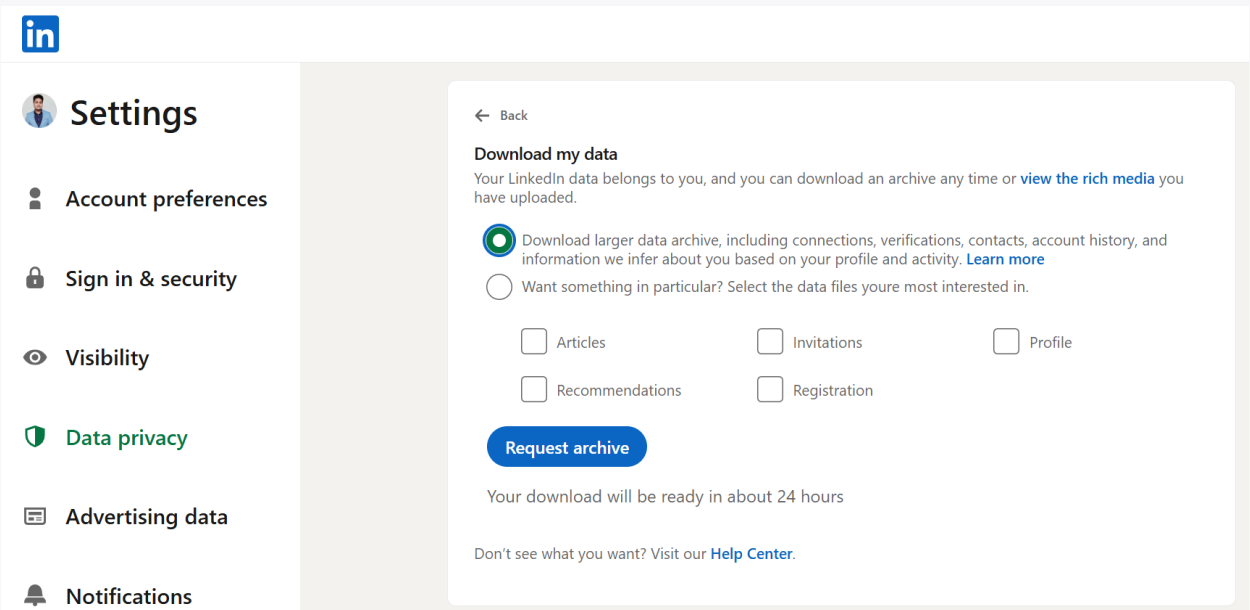
dapto

STEP 1 / EXPORT

Your LinkedIn files

Open Settings & Privacy, then Data privacy.

Choose Get a copy of your data / Download your data.



Select Download larger data archive.

Click Request archive. Open the ZIP from LinkedIn's email.

Keep these two files:

Connections.csv **Invitations.csv**

STEP 2 / JEV SETUP

Jev in Claude Code

If you completed Part 1, skip the installation commands.

Run in your terminal:

```
claude plugin marketplace add typesafe-ai/skills

claude plugin install typesafe@typesafe-ai
```

Open Claude Code and invoke the skill:

```
/typesafe:typesafe-ai
```

Create your TypeSafe API key at console.typesafe.ai.
The app will use it as TYPESAFE_API_KEY.

STEP 3 / APP SERVICES

Accounts and access

Prepare these accounts before building.

1. **GitHub and Vercel**

   Sign in to both. In Claude Code, ask: “Check my GitHub and Vercel access. Help me connect them.”

2. **Apify for extra profile details**

   Create an account. Open Console, then API & Integrations, and create a token for this app.

3. **Keys for the deployed app**

   Once Claude creates the Vercel project, add TYPESAFE_API_KEY and APIFY_TOKEN in Settings.

Keep the keys out of the prompt and GitHub repository.

dapto

STEP 4 / PLAN MODE / PROMPT 1 OF 3

The app and your data

Use the TypeSafe skill. Plan a private web app called LinkedIn Radar. Explain the plan in plain English before building. Use Next.js, TypeScript and Tailwind.

I will upload my own Connections.csv and Invitations.csv in the app. Do not expect files in this chat. Create one upload slot for each file, then separate Connections and Invitations tabs.

Detect files by their columns. Skip any Notes preamble before the Connections header. Parse quoted fields, dates and blanks. Show record counts and missing fields.

Add an editable buyer profile: my offer, target roles, industries, geography, company size and exclusions. Ask me for these values. Add a profile grid, filters and CSV export.

Use Shift+Tab for Plan mode. Paste prompts 1-3 together.

STEP 4 / PLAN MODE / PROMPT 2 OF 3

The Jev questions and shortlist

Use Jev via the official TypeSafe SDK. For connections, classify role, any industry signal and explicit exclusions using the supplied title and company text. Keep unknown and unclear answers available.

Create Tier 1 for strong matches, Tier 2 for uncertain or missing details, Tier 3 for weaker matches, and Rejected for explicit exclusions. Show the reason for each tier. Code handles dates, matching and tier rules.

For incoming invitations with a message, classify intent and stated seniority. Put empty messages in No signal. Match outgoing profile URLs to Connections in code. Unmatched status stays unknown.

Add a Classifier Studio to edit questions, answer options and cutoffs. Include Test on 25 rows. Cutoff edits reuse saved answers. Question edits require rescoring.

Use Shift+Tab for Plan mode. Paste prompts 1-3 together.

dapto

STEP 4 / PLAN MODE / PROMPT 3 OF 3

Enrichment and deployment

Add optional Apify enrichment for selected profiles. Inspect the actor’s current schema and price first. Show estimated cost and get approval before a five-profile test or larger run. Rescore using returned evidence.

Keep missing location, size, ownership, revenue and activity unknown. Never infer them from a company name. Show the evidence for each result.

Keep keys server-side. Protect the app and API routes with authentication. Keep uploaded contacts out of GitHub and logs. Send only selected scoring fields to Jev and selected URLs to Apify.

Add Run, Pause, Resume and elapsed time. Track Jev and Apify costs separately. Test with synthetic files, then deploy to Vercel from a private GitHub repo. The app sends no LinkedIn messages.

Use Shift+Tab for Plan mode. Paste prompts 1-3 together.

STEP 5 / BUILD

From the plan to your app

Review the plan, then give Claude Code this instruction.

> Build the approved plan. Test both CSV schemas and missing fields. Show me the test results, help me enter keys securely, then deploy the protected app and send me its URL.

1 Check the tests

File parsing works. Blank titles go to Review.
No-message invitations skip intent scoring.

2 Open the deployed URL

Sign in and confirm you see the two upload slots.
If a key or account is missing, resolve that first.

dapto

STEP 6 / UPLOAD

Your files go into the app

Open LinkedIn Radar. Upload the CSVs from Step 1.

csv **Connections.csv**

Drop here or choose file. This activates Connections.

csv **Invitations.csv**

Drop here or choose file. This activates Invitations.

Check record counts and missing job titles before running Jev.

Either file can work on its own. Upload both for URL matching.

dapto

STEP 7 / BUYER PROFILE

Who do you want to reach?

Enter your own criteria. This example shows how.

Offer	Operations consulting
Target roles	COO, Chief of Staff, operations leaders
Company fit	Manufacturing, distribution or logistics
Geography	United States, checked after enrichment
Exclude	Recruiters and agencies selling to me

Jev uses these criteria to judge fit.
You can change them for your own offer.

STEP 8 / SMALL TEST

25 rows before the full run

Open Classifier Studio, then choose Test on 25 rows.

Does a COO match the role?

Check that the answer follows your buyer criteria.

Does a recruiter get excluded?

Only if your exclusions explicitly include recruiters.

What happens with a blank title?

The role stays unknown and the row goes to Review.

Wrong answer? Edit the question and test again.
Wrong tier? Adjust the cutoff using the saved answers.

dapto

STEP 9 / SCORE

Your network becomes a shortlist

Run the questions you approved on the uploaded rows.

Connections

Filter by target role and tier. Inspect why each person made the shortlist.

Incoming invitations

Separate relevant messages from sales pitches and other requests. Review No signal separately.

Progress and cost

Watch completed rows, elapsed time and actual usage. Pause or resume when needed.

Jev judges the text. The app applies your shortlist rules.

STEP 10 / OPTIONAL

More detail for the best matches

Select the profiles you want Apify to enrich.

1. **Start with five profile URLs**

   Check the actor, its estimated cost and the fields it actually returns before a larger run.

2. **Add the available public details**

   Use returned location, headline, company details or activity. Leave absent details unknown.

3. **Run the second Jev pass**

   Check fit against geography and company criteria using the added evidence, then review the new tiers.

Enrich your shortlist first to control the enrichment spend.

STEP 11 / OUTREACH

The people to contact first

Filter Tier 1, inspect the evidence, then export the CSV.

EXAMPLE SHORTLIST RECORD

Role	Chief Operating Officer
Company	Industrial business
Reason	Target role and industry match
Next step	Open profile and verify fit

1 Choose the right people

Review their profiles and the reason for the match.

2 Make your message relevant

Use what you learned to write your own outreach.

The app creates the list. You send the messages yourself.

THE RESULT

A relevant list for your next outreach campaign

Your LinkedIn export supplies the contacts.
Your buyer profile defines a good match.
Jev helps you sort and review them.

Save this tutorial and start with your own export.

REFERENCE LINKS

docs.typesafe.ai/agent-skill
linkedin.com/help/linkedin/answer/a1339364
docs.apify.com/integrations/api
vercel.com/docs/environment-variables