Cadre Backend Automation Plan

Context

Cadre is a job board for VC-backed companies. The frontend (cadre-ui) is in good shape. The backend needs work: many Airtable fields are empty despite data being available, and there's no automation pipeline for ongoing enrichment.

Repos:

- matthewcmccool-cloud/cadre-ui - Next.js frontend + API routes (this repo)
- matthewcmccool-cloud/cadre-jobs-sync - Job sync engine (ATS connectors)

Airtable base ID: stored in AIRTABLE_BASE_ID env var on Vercel

---

Current State of Airtable

Table: Job Listings (~16,000+ records)

Table ID: tbl4HJr9bYCMOn2Ry

Field	Type	Status	Source
Job ID	Text	[ok] Populated	Sync
Title	Text	[ok] Populated	Sync
Companies	Linked Record -> Companies	[ok] Populated	Sync
Company	Text	[ok] Populated	Sync
Company Industry (Lookup)	Lookup	[ok] Auto from Companies	-
Investors	Lookup	[ok] Auto from Companies->VCs	-
Job URL	URL	[ok] Populated	Sync
Apply URL	URL	[ok] Populated	Sync
Raw JSON	Long Text	[ok] Populated	Sync (full ATS response)
content	Long Text	[!] Partial	Sync (HTML from ATS)
Job Description	Long Text	[!] Partial	Sync
Date Posted	Date	[ok] Populated	Sync
Last Seen	Date	[!] Partial	Sync
Location	Text	[x] Mostly empty	Extractable from Raw JSON
Function	Linked Record -> Function	[x] Mostly empty	Classifiable from title
Remote First	Checkbox	[x] Empty	Extractable from Raw JSON
Salary	Text	[x] Empty	Extractable from Raw JSON (if present)
Country	Text	[x] Empty	Derivable from location
ATS Platform (Lookup)	Lookup	[x] Mostly empty	Auto from Companies

Likely legacy fields (confirm with Matt before deleting):

Field	Why likely legacy
Matched Keyword	Old matching approach, no code references it
About	Belongs on Companies table, not Job Listings
AI Score	No code uses it; may be planned future feature
is_featured	Future monetization - not built yet
featured_tier	Future monetization - not built yet
featured_start	Future monetization - not built yet
Featured End	Future monetization - not built yet
stripe_payment_id	Future monetization - not built yet
featured_status	Future monetization - not built yet
Type (from...)	Lookup field with no visible use

Note: The featured_ and stripe_ fields may be planned features, not legacy. Don't delete without confirmation - just ignore them for now.

Table: Companies (~1,343 records)

Table ID: tbl4dA7iDr7mjF6Gt

Field	Type	Status	Source
Company	Text	[ok] Populated	Manual / scrape
URL	URL	[!] Partial	Manual / scrape
Jobs (Linked)	Linked Record -> Jobs	[ok] Auto	-
VCs (Linked)	Linked Record -> Investors	[!] Partial	Portfolio scrape
Industry (Linked)	Linked Record -> Industry	[!] Partial	-
ATS Platform	Single Select	[!] Partial	Detection needed
Jobs API URL	URL	[!] Partial	Detection needed
Last Sync	Date	[!] Partial	Sync engine
Rank	Number	[!] Partial	Manual
About	Long Text	[x] Mostly empty	Perplexity enrichment
Stage	Single Select	[x] Mostly empty	Perplexity enrichment
Total Raised	Currency	[x] Empty	Perplexity enrichment
Size	Single Select	[x] Empty	Perplexity enrichment
LinkedIn URL	URL	[x] Empty	Perplexity enrichment
Twitter URL	URL	[x] Empty	Perplexity enrichment
HQ Location	Text	[x] Empty	Perplexity enrichment
Year Founded	Number	[x] Empty	Perplexity enrichment
Status	Single Select	[x] Empty	TBD
Last Error	Text	[x] Empty	Sync error logging

Stage values: Early Stage, Mid Stage, Late Stage, Public

Size values: 1-50, 51-200, 201-1000, 1000+

Table: Investors (~201 records)

Table ID: tblH6MmoXCn3Ve0K2

Field	Type	Status	Source
Company	Text	[ok] Populated	Manual

Field	Type	Status	Source
PortCo's (Linked)	Linked Record -> Companies	[ok] Populated	Portfolio scrape
Website	URL	[ok] Populated	Manual
LinkedIn	URL	[ok] Populated	Manual
Portfolio URL	URL	[ok] Populated	Manual
URL Verified	Checkbox	[!] Partial	Manual
Last Scraped	Date	[!] Partial	Scrape endpoint
Job Listings	Lookup	Auto	-
Job Feed	URL	[x] Empty	TBD
Bio	Long Text	[x] Mostly empty	Perplexity enrichment
Location	Text	[x] Mostly empty	Perplexity enrichment
Fundraises (Linked)	Linked Record -> Fundraises	[!] Partial	-

---

Table: Function

Table ID: tbl94EXkSIEmhqyYy

Simple lookup table. Categories:

```
Solutions Engineering, Developer Relations, Revenue Operations,
BD & Partnerships, Customer Success, Product Design / UX,
Product Management, AI & Research, Engineering, Sales, Marketing,
People, Finance & Accounting, Business Operations, Legal, Other
```

Table: Industry

Simple lookup table. Field: Industry Name

Table: Fundraises

Visible in tabs but not yet explored. Linked to Investors.

Existing API Endpoints

All in cadre-ui/app/api/. Each is a Vercel serverless function (10s timeout on Hobby plan).

Endpoint	Purpose	Status
/api/backfill-functions	Classify jobs into Function categories via title regex	[ok] Working
/api/enrich-companies	Fill Stage + Size via Perplexity	[ok] Working
/api/enrich-investors	Fill Bio + Location via Perplexity	[ok] Working
/api/enrich-ats-urls	Find Jobs API URL via Perplexity	[ok] Working
/api/scrape-portfolios	Discover companies from investor portfolio pages	[ok] Working
/api/test-all	Test Airtable connectivity	Debug
/api/test-ats	Test ATS URLs	Debug
/api/test-perplexity	Test Perplexity API	Debug

All enrichment endpoints are incremental - they filter for empty fields and process in small batches (10-50

records) to stay under Vercel's timeout. They return { hasMore: true } when there are more records to process, so they need to be called repeatedly.

What Needs to Be Built

Phase 1: Backfill Job Listings from Raw JSON (No API calls needed)

New endpoint: /api/backfill-jobs

Parse the Raw JSON field (already populated for 16k+ jobs) and write back:

1. Location - Already have extraction logic in lib/airtable.ts (lines 296-328):
   - Greenhouse: rawData.offices[0].location or rawData.offices[0].name
   - Lever: rawData.location (string or object)
   - Ashby: rawData.location.city + rawData.location.country
2. Remote First - Parse from Raw JSON:
   - Greenhouse: rawData.offices containing "Remote" in name/location
   - Lever: rawData.categories.commitment === "Remote" or location contains "Remote"
   - Ashby: rawData.isRemote flag
3. Country - Derive from location string:
   - Parse city/state/country from location
   - Default "United States" for US cities, otherwise extract from string
4. Salary - Extract if present in Raw JSON:
   - Greenhouse: rawData.pay or compensation fields
   - Lever: rawData.salaryRange or rawData.categories.salary
   - Ashby: rawData.compensationTierSummary
   - Many jobs won't have this - that's expected

Implementation pattern:

```
Fetch jobs where Location = BLANK() (paginated, 100 at a time)
-> Parse Raw JSON for each
-> Extract location, remote, country, salary
-> Batch update 10 records at a time
-> 200ms delay between Airtable calls
-> Stop after 8 seconds (Vercel safety)
-> Return { updated: N, hasMore: boolean }
```

Estimated impact: Should populate Location for most of the 16k+ jobs.

Phase 2: Enhance Company Enrichment

Upgrade existing /api/enrich-companies

Currently only fills Stage and Size. Expand to also fill:

- About - 1-2 sentence company description
- HQ Location - Headquarters city
- LinkedIn URL - Company LinkedIn page
- Twitter URL - Company Twitter/X page
- Year Founded - Founding year
- Total Raised - Total funding raised

The Perplexity prompt should request all fields in one call to minimize API usage:

```
For the company [NAME] ([URL]):
1. One-sentence description
2. Headquarters city and state/country
3. LinkedIn URL
4. Twitter/X URL
5. Year founded
6. Total funding raised (USD)
7. Funding stage (Early Stage / Mid Stage / Late Stage / Public)
8. Approximate employee count (1-50 / 51-200 / 201-1000 / 1000+)

Return as JSON.
```

Only update fields that are currently empty (don't overwrite existing data).

Phase 3: Enhance ATS Detection

Upgrade existing /api/enrich-ats-urls

For companies missing ATS Platform and Jobs API URL:

1. Try known URL patterns directly (no AI needed):
   - https://boards-api.greenhouse.io/v1/boards/{slug}/jobs
   - https://api.lever.co/v0/postings/{slug}
   - https://api.ashbyhq.com/posting-api/job-board/{slug}
2. Generate slug candidates from company name (lowercase, no spaces, various formats)
3. HTTP HEAD each URL - if 200, we found it
4. Only fall back to Perplexity if direct probing fails

This is faster and cheaper than the current approach (which calls Perplexity for every company).

Also set ATS Platform field when we find the URL:

- URL contains greenhouse -> "Greenhouse"
- URL contains lever -> "Lever"
- URL contains ashby -> "Ashby"
- Otherwise -> "Custom"

Phase 4: Automation Pipeline

Goal: When data changes in Airtable, enrichment happens automatically.

Option A: GitHub Actions cron (recommended)

Create .github/workflows/enrich.yml in cadre-ui:

```
name: Daily Enrichment
on:
  schedule:
    - cron: '0 6   *'  # 6 AM UTC daily
  workflow_dispatch: # Manual trigger

jobs:
  enrich:
    runs-on: ubuntu-latest
    steps:
      - name: Backfill job fields from Raw JSON
        run: |
          for i in {1..20}; do
```

```
            RESULT=$(curl -s "${{ secrets.SITE_URL }}/api/backfill-jobs")
            HAS_MORE=$(echo $RESULT | jq -r '.hasMore')
            if [ "$HAS_MORE" != "true" ]; then break; fi
            sleep 2
          done

      - name: Backfill job functions
        run: |
          for i in {1..20}; do
            RESULT=$(curl -s "${{ secrets.SITE_URL }}/api/backfill-functions")
            HAS_MORE=$(echo $RESULT | jq -r '.hasMore')
            if [ "$HAS_MORE" != "true" ]; then break; fi
            sleep 2
          done

      - name: Enrich new companies
        run: |
          for i in {1..10}; do
            RESULT=$(curl -s "${{ secrets.SITE_URL }}/api/enrich-companies")
            HAS_MORE=$(echo $RESULT | jq -r '.hasMore')
            if [ "$HAS_MORE" != "true" ]; then break; fi
            sleep 3
          done

      - name: Detect ATS URLs
        run: |
          for i in {1..10}; do
            RESULT=$(curl -s "${{ secrets.SITE_URL }}/api/enrich-ats-urls")
            HAS_MORE=$(echo $RESULT | jq -r '.hasMore')
            if [ "$HAS_MORE" != "true" ]; then break; fi
            sleep 3
          done

      - name: Enrich investors
        run: |
          for i in {1..5}; do
            RESULT=$(curl -s "${{ secrets.SITE_URL }}/api/enrich-investors")
            HAS_MORE=$(echo $RESULT | jq -r '.hasMore')
            if [ "$HAS_MORE" != "true" ]; then break; fi
            sleep 3
          done
```

Option B: Airtable Automations (simpler but less flexible)

- Trigger: When record created/updated
- Action: Call webhook URL -> /api/enrich-companies?id=RECORD_ID
- Pro: Real-time. Con: Airtable automation limits on free plan.

Recommendation: Start with GitHub Actions cron. It's free, reliable, and works with the existing incremental batch pattern. Add Airtable automations later for real-time enrichment of individual new records.

Order of Operations

Immediate (this session)

1. Clean up legacy fields - Confirm with Matt which Job Listing fields to remove, then delete from Airtable
2. Build /api/backfill-jobs - Parse Raw JSON -> write Location, Remote First, Country, Salary back to Airtable

3. Run backfill-jobs - Execute repeatedly until all 16k+ jobs have Location populated

Next session

4. Expand /api/enrich-companies - Add About, HQ Location, LinkedIn, Twitter, Year Founded, Total Raised
5. Improve /api/enrich-ats-urls - Direct URL probing before Perplexity fallback
6. Run enrichment - Process all companies with empty fields

After that

7. Set up GitHub Actions cron - Daily automation for all enrichment endpoints
8. Integrate with cadre-jobs-sync - When sync adds new jobs, trigger backfill automatically

---

Critical Rules (from CLAUDE.md + past bugs)

1. NEVER use response.json() - Always response.text() + JSON.parse()
2. Use fetchAllAirtable for tables with 100+ records (Companies, Investors, Jobs)
3. Airtable rate limit: 5 req/sec - always add 200ms delays
4. Vercel timeout: 10 seconds on Hobby plan - stop processing at 8s
5. Batch updates: Max 10 records per Airtable PATCH request
6. Deduplicate by Job URL before creating records
7. Only update empty fields - never overwrite existing data during enrichment

Environment Variables Required

```
AIRTABLE_API_KEY     - Airtable Personal Access Token
AIRTABLE_BASE_ID     - Airtable Base ID
PERPLEXITY_API_KEY   - For AI-powered enrichment
```

All already configured on Vercel.

Airtable API Patterns

Batch update records

```
// PATCH https://api.airtable.com/v0/{baseId}/{tableId}
// Body: { records: [{ id: "recXXX", fields: { "Location": "San Francisco" } }] }
// Max 10 records per request
```

Filter for empty fields

```
filterByFormula: '{Location} = BLANK()'
filterByFormula: 'AND({Location} = BLANK(), {Raw JSON} != BLANK())'
```