A Beginner’s Guide to Building a Job Scraping Automation with n8n and AI

📘 Job Scraping Automation – Step-by-Step Build Guide

Author: Sachin Savkare
Tools Used: n8n, Google Gemini, JSearch API, Google Sheets
Goal: Automate job discovery, analysis, and recommendation without manual effort.

🧩 Overview

This workflow helps you:

- Find fresh, relevant job listings for a selected role
- Use AI to extract ATS keywords and personalized recommendations
- Organize and store all results in Google Sheets

✅ Prerequisites Checklist

- n8n Account – Sign Up (Free trial or self-hosted)
- Google Gemini API Access
- JSearch API key from RapidAPI
- Google Sheets account with a sheet for your project data
- Project portfolio data (stored in Google Sheets)
- Basic understanding of using nodes in n8n

▯ Step-by-Step Workflow Build

🔷 Step 1: Create Your n8n Workflow

1. Login to n8n
2. Click **“New Workflow”**
3. Name it: Smart Job Scraper Automation

🔷 Step 2: Manual Trigger

1. Add node: **Manual Trigger**
2. This will start the workflow manually (you can later switch to scheduled trigger)

🔷 Step 3: Set Target Role

1. Add node: **Set**
2. Add variable:
   - Name: role_name
   - Value: "Data Analyst" (or your preferred role)

🔷 Step 4: Generate Job Titles Using Gemini AI

1. Add node: **HTTP Request**
2. Connect this to **Google Gemini API**
3. Use this prompt:

pgsql

CopyEdit

List all possible job titles that a person aspiring to become a (Data Analyst) can apply for. Include every relevant variation or adjacent role such as Product Analyst, SQL Administrator, Power BI Developer, BI Analyst, etc. Return only job titles, separated by the "OR" keyword. Do not include any explanation.

4. Output should be:

sql

CopyEdit

Data Analyst OR Product Analyst OR BI Developer OR SQL Administrator ...

🔷 Step 5: Search Jobs via JSearch API

1. Add node: **HTTP Request**
2. Method: GET
3. URL: https://jsearch.p.rapidapi.com/search
4. Headers:
   - X-RapidAPI-Key: *(Your API Key)*
   - X-RapidAPI-Host: jsearch.p.rapidapi.com
5. Parameters:
   - query: Use output from Step 4
   - country: in

    - date_posted: today
    - job_requirements: more_than_3_years_experience

🔷 Step 6: Split the Job Listings

1. Add node: **Split In Batches**
2. Set it to process each job listing individually

🔷 Step 7: AI Analysis: ATS Keywords + Recommendations

1. Add node: **HTTP Request** (Gemini AI again)
2. Prompt:

csharp

CopyEdit

You are an expert in data-related hiring. Based on the given JD:

Task 1: Extract 10–15 top ATS keywords from the job description.

Task 2: Recommend the best-fit candidate profile and match it with projects from this portfolio [link to your Google Sheet].

Output in JSON format:

```
{
  "ID": "<job_id>",
  "key_words": [...],
  "talent_manager_comment": "...",
  "relevant_projects": [{"Project Name": "...", "Course": "..."}]
}
```

🔷 Step 8: (Optional) Parse AI Output

1. Add node: **Code** or **JSON Parse**
2. This ensures clean structured data for the next steps

🔷 Step 9: Merge Original Job + AI Analysis

1. Add node: **Merge**
2. Merge the original job data with Gemini AI output using:
    - Merge By: job_id (from job listing) = ID (from AI output)

🔷 Step 10: Save to Google Sheets

1. Add node: **Google Sheets**
2. Connect your Google Account
3. Choose your destination spreadsheet
4. Set operation: Append or Update
5. Map fields:
    - job_id
    - job_title
    - employer_name
    - job_apply_link
    - key_words
    - talent_manager_comment
    - relevant_projects

🎯 Final Output

Once complete:

- You click **"Execute Workflow"**
- It fetches jobs, analyzes them with AI, and updates Google Sheets
- You get:
    - Job Title + Company + Apply Link
    - ATS Keywords
    - Recommendation comment
    - Matching projects from your portfolio

📌 **Tips for Beginners**

- Keep testing each node step-by-step using n8n's **"Execute Node"** button
- Save your workflow frequently
- Use **logs and debugging** to troubleshoot JSON parsing or API errors
- You can later schedule this workflow to run **daily/weekly**