Upwork Job Scraper → AI Analysis → Airtable (n8n) — Documentation

AUTHOR - Nikhil Singh Adhikari

1) Goal

Automate this pipeline:

1. Fetch latest Upwork jobs using an Apify actor
2. Clean/rename fields inside n8n
3. Analyse each job using an LLM (Gemini/Cohere/OpenAI/Ollama)
4. Save final structured results into Airtable

Output is an Airtable table containing:

- `title`
- `description`
- `score`
- `reason`
- `url`

---

2) Final Workflow Overview (Nodes)

Schedule Trigger → HTTP Request (Apify) → Edit Fields → Analyse Job (LLM) → Structured Output Parser → Airtable: Create Record

---

3) Credentials Required

Apify

- Needs **Apify API token**
- Used in **HTTP Request** node to authenticate Apify API calls

Airtable

Use **Personal Access Token** (recommended over OAuth2).
Token must include scopes:

- `data.records:read`
- `data.records:write`
- `schema.bases:read`

Token must have access to the base:

- Base: **Upwork Automation** (or your base)

LLM Provider

- OpenAI API key (paid billing needed for stable use)
- Google Gemini API key (free tier is rate-limited)
- Cohere API key
- Ollama (local; no external billing)

---

4) Setup Steps

Step A — Apify (Job Scraper)

1. Pick an actor that works in your account:

    - Example: `jyaba/upwork-jobs-scraper`

2. Run it once inside Apify UI so it creates a dataset.
3. Copy:
    - Actor endpoint OR Dataset endpoint
    - Token from Apify settings

Common mistake fixed: using the wrong URL format (token placed incorrectly).
Correct pattern depends on endpoint, but token must be either:

- Query param `?token=YOUR_TOKEN`
  or
- Header `Authorization: Bearer YOUR_TOKEN` (preferred)

Step B — Airtable Table

1. Create a base: **Upwork Automation**
2. Create a table: **Upwork Jobs**
3. Create these fields (exact names matter if auto-map):

- `title` (single line text)
- `description` (long text)
- `score` (number)
- `reason` (long text)
- `url` (URL field)

Note: Airtable "URL field type" may appear as "URL" or as a link/icon. If you don't see URL type, use **Single line text** (works fine).

---

5) Node Configuration (What We Did)

5.1 Schedule Trigger

- Purpose: Runs workflow automatically.
- Set frequency: **[every X minutes / daily / hourly]**
- Can be paused anytime and resumed later.

5.2 HTTP Request (Apify)

- Purpose: Fetch job items.
- Method: `GET` or `POST` depending on endpoint.

Used endpoints (example):

- Get dataset items:
  - `GET https://api.apify.com/v2/datasets/[DATASET_ID]/items?token=[APIFY_TOKEN]`

Errors we fixed

- `Authorization failed / Forbidden`
  Cause: wrong token placement or wrong actor endpoint.
  Fix: use correct endpoint + valid token + correct query/header auth.
- Actor is paid
  Fix: switch to a free actor or use a dataset already created.

5.3 Edit Fields (Normalization)

- Purpose: standardize keys so the AI + Airtable step gets clean consistent fields.

We ensured each item contains:

- `title`
- `description`
- `url`
- (optional) `relativeDate`, `jobType`, etc.

Issue we fixed: `url` was coming as `undefined`
Cause: wrong field name. Upwork URL was stored under something like:

- `jobUrl`
  Fix: map correctly:
- `url = {{$json.jobUrl}}` (or the correct source key)

5.4 Analyse Job (LLM)

- Purpose: score + reason output from the job text.

Prompt output format (structured):

- score: number from 1–10
- reason: short explanation
- title: job title
- url: job url

Errors we fixed

- OpenAI model not found (example: `chatgpt-4o-latest`)
  Cause: wrong model name OR no API access.
  Fix: pick a model you actually have access to, or switch provider.
- `Insufficient quota`
  Cause: no billing / quota exceeded.
  Fix: enable billing or use another provider.

- Gemini `429 Too Many Requests`
  Cause: free tier rate limit.
  Fix: lower execution frequency, reduce items per run, or switch provider (Cohere/Ollama/OpenAI paid).

5.5 Structured Output Parser

- Purpose: convert LLM response into clean JSON fields.
- Output fields produced:
  - `title`
  - `score`
  - `reason`
  - `url`

This makes Airtable mapping simple.

5.6 Airtable — Create a record

- Purpose: Save each analysed job into Airtable.

Correct settings

- Credential: Airtable Personal Access Token
- Resource: Record
- Operation: Create
- Base: `app...` (Base ID)
- Table: **Upwork Jobs** (either by name or `tbl...` Table ID)
- Mapping mode: **Map Each Column Manually** (recommended)

Fields we send

- title → `{{$json.title}}`
- description → `{{$json.description}}`
- score → `{{$json.score}}`
- reason → `{{$json.reason}}`
- url → `{{$json.url}}`

---

6) Problems Faced + Solutions

Problem 1 — Apify “Forbidden / Auth failed”

Cause: token placed wrong / wrong endpoint.
Fix: use dataset items endpoint and attach token correctly.

Problem 2 — Actor endpoint didn’t work or actor was paid

Cause: selected actor requires paid plan.
Fix: switch to a free actor or use dataset endpoint after running once.

Problem 3 — OpenAI model not found / no access

Cause: wrong model name or no API entitlement.
Fix: choose accessible model or switch provider.

Problem 4 — “Insufficient quota”

Cause: API account has no credit/billing.
Fix: add billing or use free/local (Ollama).

Problem 5 — Gemini 429 rate limit

Cause: free tier request per minute limit.
Fix: reduce trigger frequency, reduce items per run, or use another provider.

Problem 6 — Airtable 403 “INVALID_PERMISSIONS_OR_MODEL_NOT_FOUND”

Cause: Airtable token missing scopes or base access OR wrong base/table id.
Fix: add scopes (`data.records:write`, etc.) + add base access + verify base/table IDs.

Problem 7 — "No columns found in Airtable"

Cause: Airtable table not selected correctly or schema fetch blocked.
Fix: confirm table exists, reselect Base/Table, ensure `schema.bases:read` scope.

Problem 8 — Priority field validation error

Cause: Airtable single select expected specific values but we were sending something else (Low/High).
Fix: either map values to allowed options OR remove the field.
Final fix: deleted priority field and removed it from n8n mapping.

Problem 9 — URL field type missing in Airtable UI

Cause: field type list doesn't show "URL" clearly.
Fix: use **Single line text** (stores links fine).

---

7) How to Verify It Works

1. Run workflow once manually (Execute workflow).
2. Go to Airtable base → **Upwork Jobs** table.
3. You should see new rows populated.
4. Check:
    - `title` filled
    - `score` numeric
    - `reason` text
    - `url` correct link

---

8) Production Notes (To Avoid Breaking Again)

- Keep Schedule Trigger frequency reasonable (rate limits).
- Limit number of fetched jobs per run (5–20).
- Always keep the LLM output structured.
- If using Gemini free tier: expect random blocks, don't rely on it for nonstop automation.
- Ollama is stable if your machine can run it.

---

9) Final Deliverable

A scheduled automation that collects Upwork jobs, scores them with AI, and stores them into Airtable in a clean table.

Note:-

As a fresher I apologies for mistakes and errors because this is the first time with n8n it was very helpful for experience with n8n and learning new things. I am grateful for the opportunity you give ,

Thank you