

UI Theme 01 · Operational Command Center

A live catalog scraper that turns a website into an operating dataset

A multi-agent system that indexes an ecommerce catalog, matches models, extracts variants, enriches live price and inventory data, and outputs an operations-ready workbook.

Show Results

View Pipeline

The dashboard displays the following information:

- Stage:** completed
- Progress:** 575/575
- Matched:** 575
- Variant Rows:** 10507
- Read Rate:** 33.3/min
- ETA:** -

Layer	Agents	Status	Progress
Readers	2	cached	cached/cached pages
Processors	10	completed	575/575 targets
Writer	1	completed	saved

Metric	Value
Elapsed	17m 16s
ETA	-
Processing Rate	33.3/min
Products Indexed	3502
Output	./Users/.../Catalog V7.8 - Pipeline UI FULL.xlsx

Agents:

- Reader Agents - Greek pantheon Intake layer: Hermes (completed), Iris (completed)
- Processor Agents - model matching and variant extraction: Athena, Hephaestus, Demeter, Ares, Hestia, Dionysus, Apollo, Artemis, Poseidon, Zeus (all completed)
- Writer Agent - workbook assembly: Calliope (completed)

Controls: Start, Pause, Resume, Live Code

3,502 products indexed from the site	575/575 target models matched	11,082 live variant rows generated	17.3m full production run time
--	---	--	--

Not a script. A data production line.

Instead of manual work across hundreds of models, the system divides the job into layers: readers, processors, and a workbook writer. Every layer reports status, rate, progress, and output.

01

Catalog Indexing

Two reader agents scan catalog pages, build the full product inventory, and create a clean base before processing begins.

02

Model Matching

Ten processor agents run in parallel, matching existing model targets against the live site and extracting variant data.

03

Live Price and Stock

The system goes beyond static scraping: it enriches live data, read rate, ETA, and operational fields.

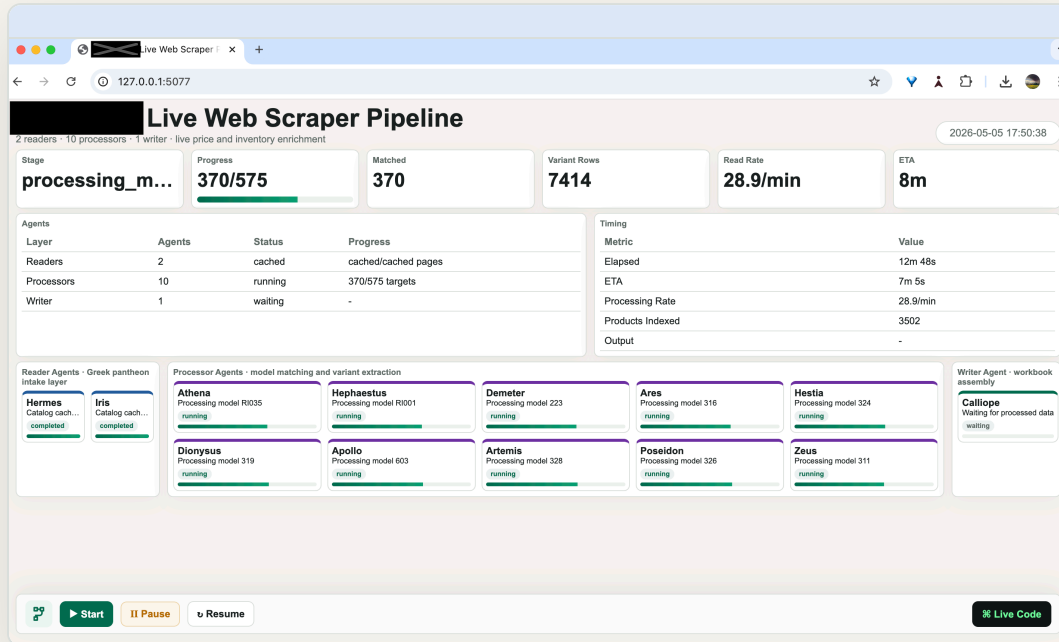
04

Workbook Output

The writer agent assembles the result into a usable Excel workbook, with expanded variant rows instead of a flat product list.

The client sees the system working.

These screenshots do not feel like a mockup. They show a real system in motion: live progress, active agents, processing rate, ETA, and a writer stage waiting for the final output.



Real-Time Control

Start, Pause, Resume, and live state tracking make long-running catalog work controllable without losing progress.



Variant Expansion

The product is not just scraping. The value is turning one product into many operationally meaningful variant rows.



Proof of Throughput

The dashboard proves execution: scanned count, matched count, generated rows, and processing speed per minute.

Technical transparency that sells trust.

The Live Code view gives buyers confidence that this is a real system with a runner, state, output, and agent architecture, not a fragile one-off automation.

```
catalog-pipeline://live-runner close

# Live Web Scraper Pipeline

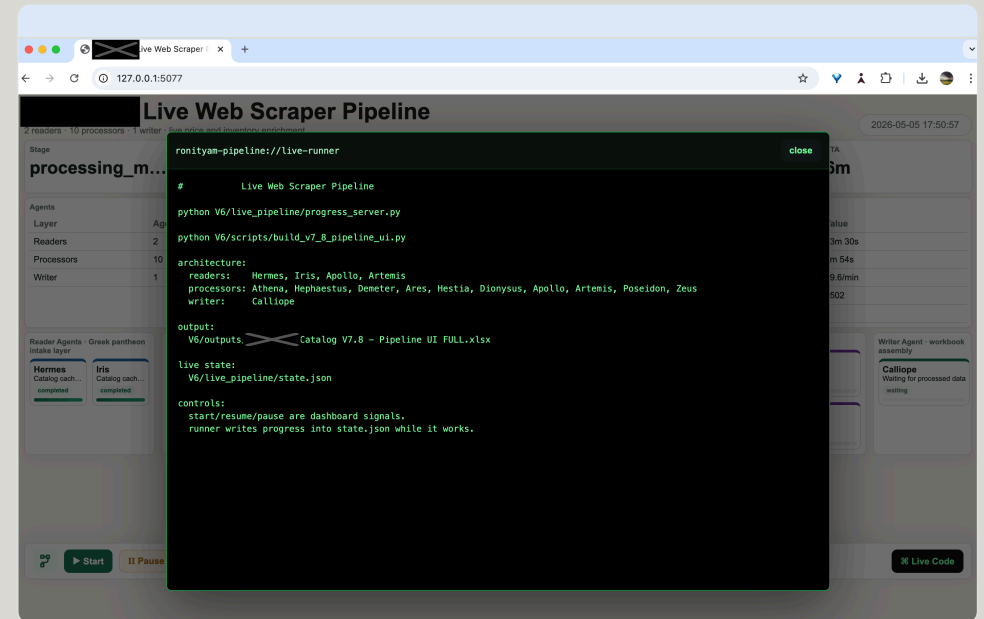
python V6/live_pipeline/progress_server.py
python V6/scripts/build_pipeline_ui.py

architecture:
  readers:    Hermes, Iris
  processors: 10 parallel model agents
  writer:     workbook assembly

output:
  anonymized_catalog_pipeline_full.xlsx

live state:
  state.json

controls:
  start / resume / pause
  runner writes progress while it works.
```



How to sell it.

The message: we are not just scraping a website. We are building a catalog intelligence layer that gives the business control over product, price, stock, and variants.

Live Catalog Intelligence Sprint

A sprint to build a live catalog pipeline that extracts the site catalog, cleans it, matches models, expands variants, and produces an operational workbook.

3.5K

indexed products

10x

parallel processors

0

missing target matches

Sales copy to use next to the screenshots

- We built a system that replaces manual catalog work with a live collection, matching, and enrichment engine.
- The system reads the site, identifies existing model targets, expands variants, and returns a work-ready Excel file.
- Instead of a product list, the client gets a live data layer: price, stock, matches, variants, and run status.
- The dashboard proves control: rate, ETA, active agents, writer status, and final output at the end of the run.
- This product fits any ecommerce business with a large catalog, scattered data, or recurring catalog updates without a manual team.