



A scheduled scraper that still runs months later

One resilience layer proven on two targets: retry with backoff, a circuit breaker, and blocks that fail loudly instead of quietly.



Sytze P.
Solution Engineer

The problem

A scraper is easy to write and hard to keep alive. The first version works on the day you build it, then the markup shifts, the host starts refusing the server's IP, and the spreadsheet quietly stops filling.

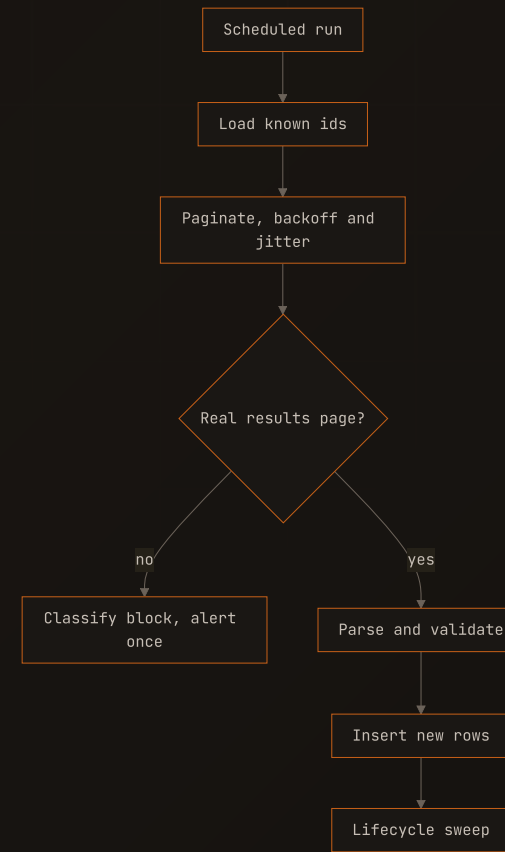
On a Dutch housing-market site the failure mode was worse than an outage. The block came back as an ordinary HTTP 200 page with no listings on it, so the run logged "no new listings", exited clean, and told nobody. A total block and a quiet market looked identical for five days.

- APPROACH

One scheduled run, end to end

Treat every failure as a state to classify, not an exception to swallow. Requests retry with exponential backoff and jitter, and a circuit breaker opens after repeated failures so a struggling target gets left alone. Any page that yields zero results is classified before it is believed: a real empty result set, a recognised block page, or markup that no longer matches.

The same core then moved to a login-gated target, where the constraints inverted. Some errors must never be retried at all, so ban signatures abort the run instead of backing off into it.



- HIGHLIGHTS

How it works

- A BLOCK NEVER LOOKS LIKE A QUIET MARKET

Zero-result pages are classified as blocked, unrecognised, or genuinely empty. Only the last one lets a run finish successfully, and the alert is throttled to one per day.

- SOME ERRORS ARE WORSE TO RETRY

On the logged-in target, detection signatures are hard non-retryable and abort the run. Backing off into a bot check deepens the flag rather than clearing it.

- A PARTIAL CRAWL NEVER WRITES A CONCLUSION

The sweep that records listings leaving the market only trusts a pass that saw the whole inventory, and skips itself when most known listings are unaccounted for.

Results & impact

four runs a day

scheduled in Docker, deployed on push to main

a block exits non-zero

a blocked run can no longer report success

no retry into a ban

detection signatures are hard non-retryable

• STACK

Tech stack

Python

async scraping pipeline

Playwright and Patchright

browser automation

Google Sheets

store and review surface

Pydantic

typed models and settings

Docker and supercronic

scheduled runs on a VPS

GitHub Actions

deploy on push to main



Sytze P.

Solution Engineer

One resilience layer proven on two targets: retry with backoff, a circuit breaker, and blocks that fail loudly instead of quietly.

