Financial News Scraper – README

Shahrouz Zada

February 11, 2025

Contents

1 Overview

This project is a comprehensive, production-grade **Financial News Scraper** designed to:

- Collect articles from multiple countries and influential news sources.
- Use proxies and `robots.txt` checks to respect each site's terms.
- Optionally harness Selenium for JavaScript-driven pages.
- Store articles in a PostgreSQL database with duplicate link avoidance.
- Provide Prometheus metrics for monitoring performance and health.

The scraping logic aims to capture key metadata—titles, publication times, article links—which can then be filtered for financial or market-related content.

2 Key Features

- **Proxy Rotation and Health Checks**: Maintains a pool of proxies, periodically re-checking dead ones.
- **Domain-Based Crawl-Delay**: Enforces rules from `robots.txt` and ensures concurrency locks at the domain level.
- **Optional Selenium Mode**: Allows scraping JavaScript-rendered pages when normal requests fail.
- **PostgreSQL Storage**: Batch inserts with `ON CONFLICT (link) DO NOTHING` to avoid duplicates.
- **Prometheus Metrics**: Exposes counters for requests, articles scraped, DB inserts, and errors.
- **Link Deduplication**: Avoids storing or re-scraping the same link multiple times in a single run.
- **Financial Keyword Filtering**: Language detection (`langdetect`) plus pre-stemmed keywords (e.g. *stock, market, economy, etc.*).

Metrics Breakdown

Metric	Use Case	Status
Deduplication	Avoid redundant articles	✅ Implemented
Proxy Health Checks	Ensure proxy reliability	✅ Robust
Crawl-Delay Compliance	Respect `robots.txt` rules	✅ Thread-safe
Selenium Concurrency	Prevent resource exhaustion	✅ Semaphore-controlled
Language Fallback	Handle detection failures	✅ Country-based

Figure 1: Metrics Breakdown

3 Architecture

High-Level Flow:

1. **Proxy Manager** selects an available proxy or marks it dead after failures.
2. **Domain Lock** ensures we respect each site's crawl-delay and do not saturate it with parallel requests.
3. **Scrape Pipeline** includes:
   - *Selenium* (optional) for JS-heavy sites.
   - *Deduplication* across multiple pages (saves memory, avoids duplicates).
4. **Data Storage**:
   - Insert articles into a *PostgreSQL* database.
   - Enforce *unique link* constraints.
   - Use JSON output as a fallback or for local debugging.
5. **Filtering**:
   - Language detection with `langdetect`.
   - Simple stemming-based matching of financial keywords.

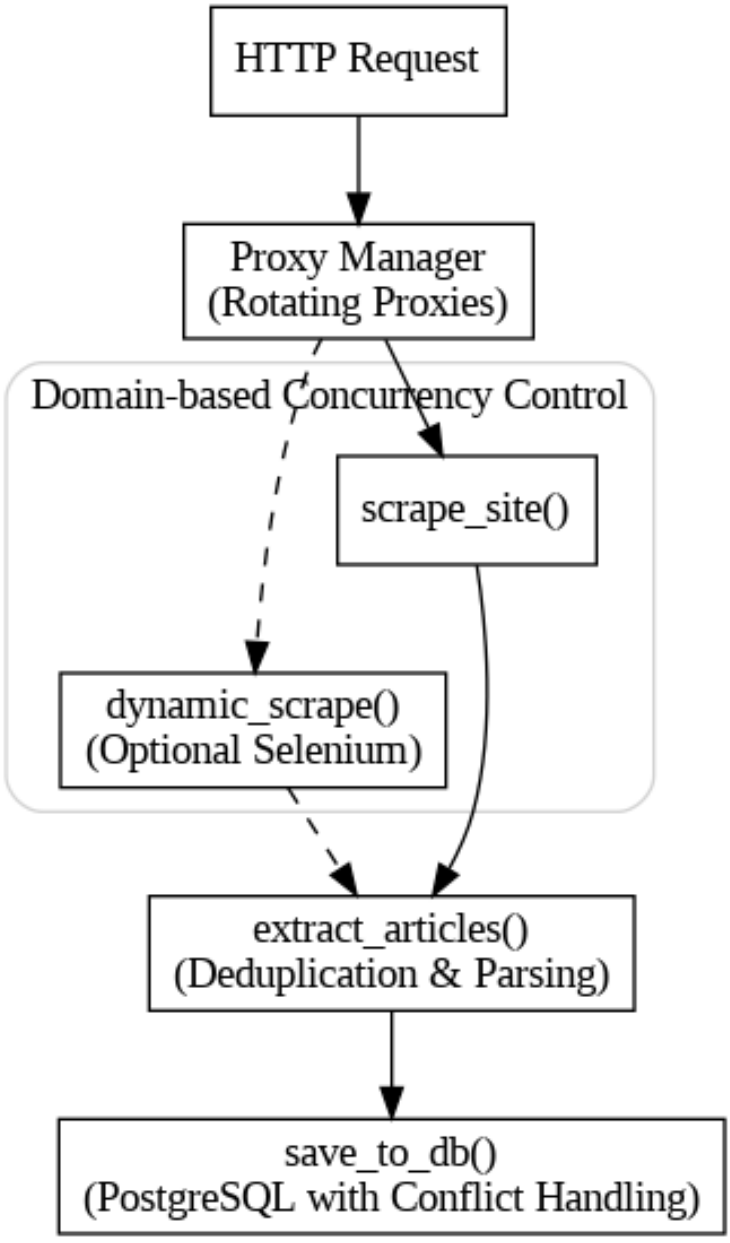

Figure 2: High-level view of the data flow

4 Prerequisites

- **Python 3.8+** with *pip* or *conda* for package management.
- **PostgreSQL** (local or remote). The code relies on `DATABASE_URL`.
- **Proxies** (optional, if not using direct requests).
- **Selenium** and **ChromeDriver** (only if `USE_SELENIUM = True`).

5 Installation & Setup

1. **Clone** this repository or copy the code base locally.
2. **Create a virtual environment** (recommended):

   ```
   python3 -m venv venv
   source venv/bin/activate
   ```

3. **Install dependencies**:

   ```
   pip install -r requirements.txt
   ```

4. **Set up PostgreSQL**:
   - Ensure `DATABASE_URL` is defined. E.g.:

     ```
     export DATABASE_URL="postgres://user:pass@localhost:5432/dbname"
     ```

6 Configuration

This project relies on environment variables and in-code toggles for runtime options:

- `DATABASE_URL`: Connection string for PostgreSQL.
- `PROXY_POOL`: Comma-separated proxies (e.g. `"http://proxy1:8080,http://proxy2:8080"`).
- `USE_SELENIUM`: Set *True* if you need JS rendering (requires ChromeDriver).
- `logging.basicConfig(...)`: For log levels (DEBUG, INFO, etc.).

7 Running the Scraper

1. **Ensure** your environment variables are correctly set (especially `DATABASE_URL`).
2. **Run the main script**:

   ```
   python main.py
   ```

3. **Monitor logs** in `scraper.log`.
4. **Check Prometheus metrics** at `localhost:8000/metrics`.

8 Filtering Logic

- Language is detected via `langdetect`.
- Titles are tokenized with regex (`\b\w+\b`).
- A language-specific stemmer (English or French) is applied to the tokens.
- We check if the stemmed tokens match a list of *pre-stemmed* financial keywords (e.g. *market, econom, bours, etc.*).
- If matches exist, the article is considered relevant and included in the final output.

9 Prometheus Metrics

We expose several counters at `http://localhost:8000/metrics`:

- `scraper_requests_total`: total HTTP requests.
- `scraper_articles_scraped`: total articles extracted from pages.
- `scraper_articles_saved`: total articles saved to the database.
- `scraper_errors`: total errors raised during scraping or DB operations.

10 Common Pitfalls & Troubleshooting

- **robots.txt Blocking**: If `can_scrape(url)` is *False*, the scraper automatically skips that domain.
- **Proxy Failures**: If all proxies fail, you may see repeated timeouts. Check logs for *"Marked proxy as dead"* messages.
- **Selenium Issues**: Make sure ChromeDriver is installed and in your `PATH` if `USE_SELENIUM = True`.
- **Database**: Ensure the `DATABASE_URL` is correct. A misconfigured environment variable results in data not being saved.

11 Contributing

Contributions are welcome!

1. **Fork** the repository and create a new feature branch.
2. Implement changes or add new countries/keywords.
3. Submit a pull request for review.

Please maintain code style (PEP 8) and update docstrings or in-line comments where needed.

12 License

No license has been specified for this project at present. For inquiries about usage or distribution, please contact the repository owners or maintainers directly.