Openpricengine API Methodologies v0.3

2024

PREPARED BY

Tony Mudau

openpricengine.com

Introduction

Openpricengine is a powerful resource offering real-time and historical price data across various sectors, including Food and Groceries, Restaurants, and Energy prices on a global scale. This document details the methodologies and technological frameworks employed to ensure the accuracy, reliability, and efficiency of our price data API. This document will detail our collection, storage and provision methodologies to ensure scalable and accurate data collection.

Data Collection Methodologies

Data Scraping

Our data collection process primarily involves web scraping using advanced Python tools. We use custom scripts designed to efficiently and responsibly scrape data from various websites. Our approach ensures minimal impact on the functionality and performance of the sites we interact with, aligning our practices with ethical web scraping standards similar to those employed by major search engines like Google. The following tools and technologies form the backbone of our data scraping operations:

- **Python**: The core programming language used for writing our data scraping scripts due to its simplicity and extensive library support.
- **Playwright**: A Node.js library used to automate browsers and facilitate the scraping of dynamic web pages. Playwright ensures we can access and extract data from complex websites efficiently.

Technology Stack for Data Collection

Our technology stack for data collection includes:

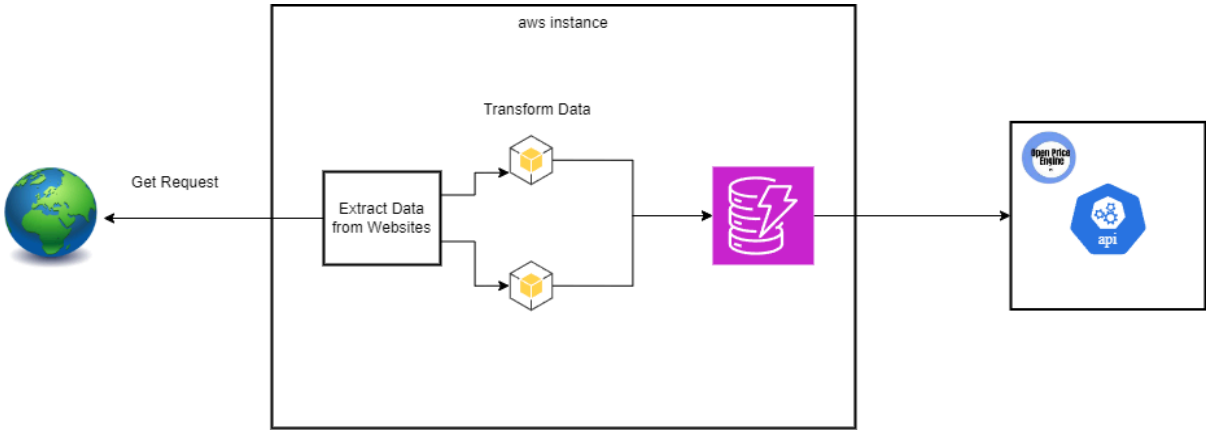
- **Linux**: Our servers run on Linux, providing a stable and secure environment for our operations.
- **AWS**: Amazon Web Services (AWS) offers scalable cloud infrastructure to support our data scraping and processing needs.
- **Pandas**: A powerful data manipulation and analysis library for Python, used to process and clean the scraped data.

Ethical and Non-Disruptive Practices

Our web scraping methodology is designed to be both ethical and non-disruptive. We adhere to best practices to ensure that our data collection activities do not negatively impact the websites we scrape. These practices include:

- **Respecting Robots.txt**: We always check and respect the `robots.txt` file of websites to honour their data collection policies.
- **Rate Limiting**: Implementing rate limiting to ensure that our scraping requests are spread out over time, reducing the load on target servers.
- **Politeness Policies**: Our scripts include delays between requests to avoid overwhelming the target websites, ensuring a minimal footprint.

Basic Architecture Diagram



Benefits of Our Web Scraping Approach

1. **Efficiency**:
    - Our custom scripts are tailored to collect data quickly and efficiently, allowing us to update our databases in real-time and provide the most current price information to our users.
2. **Accuracy**:
    - By using tools like Playwright, we can interact with web pages just as a human user would, ensuring that we capture accurate and relevant data, especially from dynamic content.
3. **Non-Disruptive**:
    - Our approach is designed to be as non-intrusive as possible, similar to how search engines crawl the web. This ensures that we do not interfere with the normal operation of the websites we scrape.
4. **Scalability**:
    - Leveraging AWS infrastructure, our web scraping operations are highly scalable. We can handle large volumes of data and scale our operations based on demand, ensuring consistent performance and availability.
5. **Compliance**:
    - We stay informed about legal and ethical considerations surrounding web scraping and ensure that our practices comply with relevant regulations and industry standards.

Data Storage Framework

Storage Solutions

To manage and store the vast amounts of data collected, we utilise the following storage solutions:

- **DynamoDB**: AWS's NoSQL database service is used for storing real-time data due to its low latency and scalability.
- **Parquet**: An open-source columnar storage format optimised for use with big data processing frameworks. Parquet is employed for storing historical data to ensure efficient querying and retrieve

API Framework

API Development and Deployment

Our API, designed to provide seamless access to our price data, is built and deployed using the following technologies:

- **FastAPI**: A modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints.
- **Uvicorn**: A lightning-fast ASGI server implementation, used to run our FastAPI application in a production environment.
- **AWS**: Our entire API infrastructure is hosted on AWS, ensuring reliability, scalability, and high availability.

The Team

Openpricengine is driven by a dedicated and diverse team of professionals who bring together their expertise to deliver a comprehensive and reliable price data API. Each member of our team plays a crucial role in maintaining the quality and integrity of our services.

Full-time Back-End Developer

Our back-end developer is at the heart of Openpricengine's technical operations. This individual is responsible for:

- **Maintenance of Data**: Ensuring the accuracy, reliability, and security of our extensive databases.
- **API Versioning**: Managing and updating the API versions to accommodate new features and improvements.
- **Documentation**: Creating and maintaining thorough documentation to support users in integrating and utilising our API effectively.

Tony Mudau

Ad-Hoc Developer

The ad-hoc developer is pivotal in expanding our data sources. Their key responsibilities include:

- **Global Data Collection**: Identifying and incorporating new data sources from around the world to enhance the comprehensiveness of our database.
- **Data Integration**: Ensuring the seamless integration of new data into our existing systems.

Suhail Bashir

Front-End Developer

Our front-end developer ensures that Openpricengine is not only functional but also visually appealing and user-friendly. Their contributions include:

- **Professional Interface**: Designing and maintaining a professional, intuitive, and engaging interface for users to interact with our API and data products.
- **User Experience**: Enhancing the overall user experience through thoughtful design and responsive features.

TAWANDA MATEWU

Business Analyst

The business analyst bridges the gap between our technical team and prospective clients. Their role encompasses:

- **Client Communication**: Engaging with potential clients to understand their needs and demonstrate how Openpricengine can meet them.
- **Market Research**: Exploring new data sources and identifying data products that are in demand by clients, ensuring our offerings remain relevant and valuable.

Tinashe (Duncan) Machiwenyika

Researcher

Our researcher, who is completing a Masters in Economics, adds a critical layer of insight and analysis to our data. Their responsibilities include:

- **Auxiliary Research**: Conducting supplementary research to support the data we collect, providing deeper insights and context.
- **Material Output**: Producing various research outputs, such as reports and analyses, that leverage our data to offer added value to our clients.

Together, our team combines their unique skills and expertise to ensure that Openpricengine remains a leading provider of real-time and historical price data. Their dedication and collaboration are the foundation of our success, enabling us to deliver high-quality services to our clients.

Lydia Raphela

Conclusion

Openpricengine leverages a robust and efficient stack of technologies to provide accurate and timely price data across various sectors. Our methodologies and frameworks are designed to ensure the highest standards of data integrity and performance, making Openpricengine a trusted resource for real-time and historical price data.