Raw Data Ingestion Service

Overview

This FastAPI application exposes endpoints for fetching, validating, and persisting raw market, on-chain, social sentiment, and news data into Parquet files (local or S3). It performs:

1. **Adapter Fetch & Normalize**
2. **Schema Validation** (raises HTTP 422 on mismatch)
3. **Atomic Parquet Write** with partitioning
4. **Observability** via /metrics and /health

Endpoints

Health & Metadata

- **GET** /
  - { "service": "raw-data-ingestion", "version": "1.0.0" }
- **GET** /health
  - { "status": "ok" }
- **GET** /metrics
  - Exposes Prometheus metrics (e.g. parquet_writes_total, parquet_write_errors_total, parquet_write_latency_seconds).

Market Data

- **POST** /ingest/market/{exchange}
  - **Body**:
    - {
      "symbol": "BTC-USDT",
      "granularity": "1m"
      }
  - **Success** (200 OK):
    - { "status": "ok", "path": "data_lake/market/exchange=binance/symbol=BTC-USDT/.../part-*.parquet" }
  - **No Data** (200 OK):
    - { "status": "no_data", "path": null }
  - **Schema Error** (422 Unprocessable Entity):
    - { "detail": "Missing columns: [...]" }
  - **Write Failure** (500 Internal Server Error):
    - { "detail": "Write failed: <error message>" }

On-Chain Data

- **POST** /ingest/onchain/{source}
  Path param: source = glassnode or covalent
  - **Body**:
    - {
      "source": "glassnode",
      "chain_id": 1,
      "symbol": "BTC",
      "metric": "exchange_netflow",
      "days": 1
      }
  - **Success** (200 OK):
    - { "status": "ok", "path": "data_lake/market/exchange=binance/symbol=BTC-USDT/.../part-*.parquet" }
  - **No Data** (200 OK):
    - { "status": "no_data", "path": null }
  - **Schema Error** (422 Unprocessable Entity):
    - { "detail": "Missing columns: [...]" }
  - **Write Failure** (500 Internal Server Error):
    - { "detail": "Write failed: <error message>" }

Social Sentiment

- **POST** /ingest/social/{platform}
  Path param: platform = twitter or reddit
  - **Body**:
    - {
      "platform": "twitter",
      "query": "BTC OR ETH",
      "since": "2025-08-01T00:00:00Z",
      "until": "2025-08-02T00:00:00Z",
      "max_results": 50
      }
  - **Success** (200 OK):
    - { "status": "ok", "path": "data_lake/market/exchange=binance/symbol=BTC-USDT/.../part-*.parquet" }
  - **No Data** (200 OK):
    - { "status": "no_data", "path": null }
  - **Schema Error** (422 Unprocessable Entity):
    - { "detail": "Missing columns: [...]" }
  - **Write Failure** (500 Internal Server Error):
    - { "detail": "Write failed: <error message>" }

News

- **POST** /ingest/news
  - **Body**:
    - {
      "source_type": "api", // or "rss"
      "feed_url": "https://example.com/rss", // only needed for rss
      "category": "crypto" // only needed for api
      }
  - **Success** (200 OK):
    - { "status": "ok", "path": "data_lake/market/exchange=binance/symbol=BTC-USDT/.../part-*.parquet" }
  - **No Data** (200 OK):
    - { "status": "no_data", "path": null }
  - **Schema Error** (422 Unprocessable Entity):
    - { "detail": "Missing columns: [...]" }
  - **Write Failure** (500 Internal Server Error):
    - { "detail": "Write failed: <error message>" }
- **GET** /v1/news/rss
  - feed_url=https://example.com/rss
  - Returns a NewsEnvelope directly without persistence.

Parquet Schema

All writes adhere to these canonical schemas (with UTC timestamps):

Domain	Columns
Market	timestamp, symbol, open, high, low, close, volume
On-Chain	timestamp, source, asset, metric, value
Social	ts, source, id, user, text, sentiment_score
News	published_at, source_type, source_name, title, url, author, summary

Partitioning is by key fields plus year, month, day derived from the timestamp.

Running Locally

1. **Install**
    a. **pip install -r requirements.txt**
2. **Env Vars**
    a. export AWS_ACCESS_KEY_ID=...
    b. export AWS_SECRET_ACCESS_KEY=...
3. **Start**
    a. uvicorn app.ingestion_service.main:app --reload --host 0.0.0.0 --port 8000
4. **Docs**
    a. Visit http://localhost:8000/docs for the OpenAPI UI.