ETL Module Detailed Points — Financial Analyst Copilot (AI-Powered RAG System)

This document summarizes all major steps in etl_sec_ingestor.py, explaining both what happens and why each step is important.

1■■ Purpose of the File

What it does:

Automates the process of downloading (Extract), cleaning (Transform), and saving (Load) SEC filings such as 10-K and 10-Q.

Why:

The ETL process creates structured, high-quality text data from real financial filings. This data powers the AI Copilot's analysis and retrieval engine.

2■■ Imports and Folder Setup

What:

os, requests, time, BeautifulSoup — standard Python libraries.

Why:

• os: For handling file paths and directories.

• requests: To make HTTP calls and download SEC filings.

• time: To manage delays between API calls (avoiding rate limits).

• BeautifulSoup: To parse and clean HTML content into readable text.

• Folder structure separates raw HTML and processed text, keeping data clean and organized.

3■■ SEC Configuration

What:

SEC_BASE_URL = "https://data.sec.gov/submissions/"

HEADERS = {"User-Agent": "AbhyudayaLohani-FinancialAnalystCopilot (abhyudaya.lohani@example.com)"}

Why:

• The SEC blocks requests without a valid User-Agent identifying the requester.

• The base URL provides access to company submission data (filing history, metadata, etc.).

• Including an identifiable User-Agent ensures compliance with SEC's access rules and prevents HTTP 403 errors.

4■■ Extract: Fetch Filings Metadata

What:

Uses the company's ticker symbol (e.g., AAPL), fetches metadata from the SEC JSON API, and extracts accession numbers.

Why:

• The metadata file tells which filings exist and where they are stored.

• Accession numbers are needed to access the actual 10-K/10-Q documents.

• The count parameter lets you control how many filings you want (useful for testing or scaling later).

5■■ Identify Filing URLs

What:

Parses the SEC filing directory and looks for .htm files (actual filings).

Why:

• Each filing folder can contain many files like exhibits, tables, and XMLs.

• The script must dynamically identify the main filing HTML file.

• BeautifulSoup allows structured extraction of hyperlinks from the SEC's HTML index pages.

6■■ Retry Logic and Downloading

What:

Retries the HTTP requests 3 times before giving up.

Why:

• The SEC enforces request rate limits, and sometimes requests can time out.

• Retrying up to 3 times increases reliability without overloading the server.

• time.sleep(2) pauses between retries to prevent being temporarily blocked.

• Successfully fetched HTML files are stored in data/raw/ for transparency and future reprocessing.

7■■ Transform: Clean Filings

What:

BeautifulSoup removes all unnecessary markup, leaving readable plain text.

Why:

• SEC filings are cluttered with HTML tags, inline styles, and scripts.

• BeautifulSoup removes all unnecessary markup, leaving readable plain text.

• The text is normalized (extra spaces removed).

• The cleaned text is saved in data/processed/, ready for NLP and AI pipelines.

8■■ Load: Save and Log Processed Data

What:

filename = os.path.basename(file_path).replace(".html", ".txt")

Why:

• Keeps naming consistent between raw and cleaned files using accession numbers.

• Ensures easy traceability — each processed text file corresponds to one raw HTML.

• Logs outputs clearly for debugging and reproducibility.

9■■ Main Pipeline Execution

What:

Combines all steps (Extract → Transform → Load) into one smooth function.

Why:

• Makes it easy to run for any company by changing the ticker symbol.

• Simplifies future automation when processing multiple tickers or datasets.

10■■ Entry Point

What:

if __name__ == "__main__":

run_pipeline("AAPL", count=2)

Why:

• Enables standalone execution using python etl_sec_ingestor.py.

• Defaults to fetching Apple (AAPL) filings for testing.

• Keeps the script modular — it can also be imported by other Python files without running automatically.

11■■ Folder Output Example

data/

■■■ raw/

■ ■■■ AAPL_000205091225000008.html

■ ■■■ AAPL_000163198225000009.html

■■■ processed/

■■■ AAPL_000205091225000008.txt

■■■ AAPL_000163198225000009.txt

Why:

• Having separate folders for raw and processed files maintains a clear ETL structure.

• Makes debugging and verification easier — you can check what was downloaded and how it was cleaned.

• Keeps the dataset well-organized for later embedding and retrieval stages.

12■■ Why This ETL Module Is Important

Why:

• Automates real-world financial data ingestion directly from the SEC.

• Converts messy, unstructured HTML filings into clean, machine-readable text.

• Builds a foundation for later project stages:

- Embedding & Indexing (turning text into vectors)

- RAG Pipeline (connecting LLM + retrieval)

- Evaluation & Guardrails (factuality, hallucination control)

- Streamlit UI (user-friendly financial analyst interface)

• Eliminates manual data gathering and ensures data consistency and reproducibility.

Summary:

etl_sec_ingestor.py is the core data ingestion engine of your Financial Analyst Copilot. It downloads, cleans, and organizes SEC filings automatically — transforming raw HTML into structured, AI-ready text for intelligent financial analysis.