Unified Security Master

March 2023

Table of Contents

1 Unified Security Master

Security master is used in a variety of financial services applications. Purpose of this exercise is as follows:

- Familiarize trainees with basic Securities domain through using Securities master data attributes.
- Learn to use data in (.csv) files and move them to databases.
- Enrich data by adding any missing attribute for classifications.
- Resolving data issues that will arise from using public data from multiple sources.
- Resolve issues around keeping data current.

1.1 Objective

Several applications in financial services viz Asset Allocation or Portfolio management system require Master Data. One such master data is the Security Master.

We are required to build a Security Master and maintain the data. Other applications will be able to consume this data, which will be exposed as APIs to them.

Diverse types of securities exist. For this application we will only consider Equity (stock) securities to exist (Equity Master). We will use public sources to build a Unified Security Master for equity stocks trading in Nasdaq, LSEG and NSE exchanges.

1.2 Features:

- Securities listed on these exchanges will be extracted and aggregated into a unified data store. The application will provide a User interface for this data. The application will also expose the data as APIs to be consumed by other applications.
- Maintaining the data set will have the following considerations:
  - Listed companies' data in stock exchanges - Nasdaq, LSEG and NSE to be aggregated into a single Master.
  - Application should be capable of reflecting changes (new company listings, change of Symbols, delisting of companies). This would require the data from Exchanges to be extracted periodically (as CSV files) and uploaded into system. It is not an one-time extract. System design to keep this in mind.
- Securities data should also have an Common Industry classification. Exchanges do provide a base classification. Team has to ensure the classification used is applicable for all Exchanges. One such classification is GICS (GLOBAL INDUSTRY CLASSIFICATION STANDARD) which is provided by MSCI. Guiding Principles and Methodology for GICS – read here. List of GICS codes – read here.
- During aggregation, data should not have duplicates. However, if a company is listed on more than one exchange, the symbol will appear under respective Exchanges. For example, if Apple is listed in both NASDAQ and LSEG stock exchanges, it may appear once under NASDAQ and once under LSEG under respective symbols.
- Strategy to keep Data current should consider real-time or batch updates based on available interfaces to the Sources of data. For example, the application can retrieve data using APIs (where exposed by the Exchange). If data is extracted from their Portals (downloaded as .CSV), it will have to be updated as a Batch routine.
- Unified Securities master will have Stock Exchange, Security Symbol, Security Series, Security Description, Security Code, Security ID (eg ISIN in NSE), Country, Currency (USD, INR, GBP).
- These above fields should not be empty, and the Application should warn if any Security does not have these values.

- Application should support for corrections to non-key values or additional information to enrich the data, using the User Interface.
- Application should support refresh of data on a weekly or adhoc basis.
- NSE data can be downloaded from their webite.
- Identify public sources for Nasdaq and London Stock Exchange traded securities. For the purposes of MVP, a sample data set from those exchanges will suffice (25 securities each).
- Application will provide Query feature for the Security Master. Users will be allowed to Query by one of the following combinations:
  - Query using fields Stock Symbol: Query will return 1 row if found (Eg. (‘AAPL') or ('RELIANCE')
  - Query using fields Security ID ('ISIN'): Query will return 1 row. Eg. ('LSEG','GB00BD97ND60')

GET /stocks/:isin

Parameters:

:isin – the ISIN of the stock to retrieve information for.

Response:

The response body will be a JSON object with following fields:

- isin – the ISIN of the stock.
- name – the name of the stock.
- symbol – the ticker symbol of the stock.
- exchange – the stock exchange where the stock is traded.
- currency – the currency in which the stock is traded.
- sector – the sector this company belongs to
- industry – the industry this company belongs to
- <<See appendix for full information>>

1.3 Non-Functional Requirements

- There are ~8000 stocks listed in US exchanges; ~4800 in LSEG;
- Application user interface should be responsive with no visible lag.
- APIs exposing the data set should return result set in less than 2 seconds
- Unit tests should be created in Junit or Nunit
- Code Quality – >70% coverage
- Teams should use a Code quality tool (eg. Sonarlint) and verify there are no ‘Blocker' or ‘Critical' errors.
- Services should write to a log file using a logging framework such as Log4J or Logstash (ELK).

1.4 Team Deliverables

- Data Model for Security master

- UI wireframes (will be completed prior to development of UI)
- Data Flow diagram
- High level architecture diagram
- Interfaces for each Exchange (source data format may be different). Initial scope for MVP is for 2 exchanges (NSE and Nasdaq).
- API Contracts (exposed for other applications)
- User interface to view Security Master.
  - Ability to view Securities by Sector, Industry (GICS classification).

1.5 Assumptions

- We will assume Application users are all internal users and have been authenticated through an existing AD.
- Teams will follow Agile Scrum and iterate in 2-week sprints.
- Demo to Business stakeholders (ASDMs / Project managers) after every Sprint.
- Demo’s will be presented by 2 members. It will be recorded by the team.

1.6 Tech Stack

- .NET or Java for Backend services
- Angular or React for UI
- Postgres or MySQL (H2 for development)
- Messaging – ActiveMQ
- Code Quality – Sonarlint / Sonarqube
- Code Repository – TBD (will be provided)
- Testing – Junit / Nunit;
- UI Testing – Selenium

thank you

www.hexaware.com