Senior Elixir Dev Exercise #1

For	Elixir Dev
Exposure	External
Type	Senior Elixir Dev
Released	Oct 3rd 2021
Authority	Infigonia Group. FZ-LLC
Team	Hiring
Owner	Pulkit Agrawal
Contributors	Pulkit Agrawal, CTO
Contact	pulkit@infigonia.com

Introduction

Code an async and distributed application for file processing.

This project should be written in a way that it can be deployed to many machines for parallel processing. Each machine will perform non-duplicate tasks in parallel.

Use any third party managed services or open source libraries.

Problem Statement

Process 1:

Fetch X numbers of CSV/Excel files from X number of API endpoints. System should allow to add more CSV sources for downloading.

Each CSV will be downloaded daily after a specific time of the day.

Each CSV can have different column names, different numbers of columns but three common columns “revenue”(decimal value) , currency(currency code like USD, EURO etc.), and date in each row.

Note: Format of one CSV will remain the same from one source.

After fetching CSV, upload it to the S3/local directory

Process 2:

Pull CSV files stored on S3 (from process 1) and process it in the database table with following changes.

1. Fetch the latest currency exchange rate from the DB table(from Process 3) by filtering on the specific currency and multiplying it with the “revenue” value for each row in CSV.
2. Mapping CSV columns to the database table columns.

Process 3:

Code to fetch currency exchange rate for USD conversion from some API(no need to define this API) and store it in the database table. Update exchange rates on a daily basis.

Design Principle

- Single project
- Fault tolerant (If process fails then it should get restarted)
- Distributed
- Data Integrity (Data after processing in DB table must not be incomplete, duplicate or corrupt)