README

Setup

Setting this up would require some preliminary tasks that need to be done first.

First, we need to download the chrome driver and browser and install it in the current directory of the folder

> **Linux chrome browser and chromedriver**
>
> Chrome Browser
> Chrome Driver

Also please note that this was run and tested with the windows subsystem for linux (WSL), so at the very least, running this would require WSL or a linux OS. But since the source code was written in the windows OS, we first need to convert the necessary files into the Unix format using the `dos2unix` tool

```
dos2unix *.sh setupDB.sql ./dags/* ./details/*
```

Once we have completed downloading and converting files, the following are the prerequisites needed for this codebase

1. python Airflow
2. Postgresql
3. python pandas
    - also download auxilliary modules to read and write parquet files
4. python selenium

Once we have those modules/softwares available, we can then proceed.

> **Danger**
>
> For WSL, there is a need to install a lot of packages such as libasound2, a good way to determine if all required packages are installed is if WSL can now run and create an instance of google browser

To setup the environment for downloading files we need to run the following commands in the shell

```
source env.sh
psql -U postgres -h localhost -a -f setupDB.sql
sh af_user.sh
```

This would do the following

1. Set environment variables as well as create the folders where the downloaded items will be stored
2. Setup the postgres user for the airflow database
3. setup an admin for airflow

Once that's done we can now actually run everything

Runtime

Running airflow DAGs can be done however one wants, but personally it is best if we can monitor them while operating. So for my approach I do the following

First, run the airflow scheduler and webserver with

```
airflow standalone
```

Log in with the credentials set up by `af_user.sh`. Feel free to change those credentials as one wishes

- Username: `andy`
- password: `pass`

Then unpause the DAG named `SGX-Derivatives`. There should only be one DAG present in the DAG list.

The DAG would open many instances of the chrome browser because by default, the DAG will download the historical data in the website. To disable this modify `config.json` to be

```
{
  "test": false,
  "dl_history": false,
  "redownload_failures": false
}
```

Note that if `dl_history` is false, then automatically `redownload_failures` will also be false

After the DAG has finished downloading, as an intermediary, the downloads will be in the `diels` folder. But if the dag is finished, it is unlikely that the files would still be there, it would instead be inside the `archive` folder.

You can track which downloads have been successful and which aren't by checking the `details/tracker.parquet`. Failed downloads will be indicated by that tracker as well as use it to redownload failures should it ever be signified in `config.json`

Logs will be available through the logs folder automatically created by airflow. The custom logs that I have made will be stored there as well

> **Note**
>
> `TickData_structure.dat` and `TC_structure.dat` will be prepended by their dates, e.g. the Tick Data Structure and TC Structure of March 9 will be
>
> ```
> TickData_structure_20240309.dat
> TC_structure_20240309.dat
> ```

As a courtesy, the submission should already have a demo runtime already done as a demonstration, I have removed the chrome browser and driver as it takes up too much memory to download, feel free to explore the directory.