Batch company pipeline

Context

We want to download and parse data coming from the Sirene Database **periodically** with the latest up-to-date information.

The Sirene Database is the open source database of all French companies declared in the commercial database given by the French government. All companies with a commercial activity are available, even those that were closed a long time ago.

The process will be to download the following files : **StockEtablissement** and **StockUniteLegale** from the government open data website.

These data are used by our data scientists to identify the frequency with which a company closes and opens a same establishment and thus to define a level of risk. SIRETs can indeed be closed and opened a large number of times within a single SIREN. This can be used by ghost companies to continue their fraudulent activities.

Data

The data you will need to download periodically is available here and here.

NB: Here, you can notice that you download the data from a shared Google Drive folder (owned by Trustpair) and not directly from the Sirene database website. This is because we had to manipulate a bit the raw files from the Sirene Database to design a sensible case study.

The data is composed of two distinct files that we describe below:

StockEtablissement file:

- Contains informations relatives to a physical establishment of a company
- They are identified via the SIRET field
- Description of the data can be found here. In this file there is other data, take care only on the schema given below to understand the meaning behind those labels.
- Schema of the file

```
root
 |-- nic: string (nullable = true)
 |-- siret: string (nullable = true)
 |-- dateFin: string (nullable = true)
 |-- dateDebut: string (nullable = true)
 |-- etatAdministratifEtablissement: string (nullable = true)
 |-- denominationUsuelleEtablissement: string (nullable = true)
 |-- activitePrincipaleEtablissement: string (nullable = true)
 |-- nomenclatureActivitePrincipaleEtablissement: string (nullable = true)
 |-- caractereEmployeurEtablissement: string (nullable = true)
```

StockUniteLegale file:

- Contains informations to the company itself as a juridical entity
- They are identified via the SIREN field
- Description of the data can be found here. In this file there is other data, take care only on the schema given below to understand the meaning behind those labels.
- Schema of the file

```
root
 |-- siren: string (nullable = true)
 |-- dateFin: string (nullable = true)
 |-- dateDebut: string (nullable = true)
 |-- etatAdministratifUniteLegale: string (nullable = true)
 |-- nomUniteLegale: string (nullable = true)
 |-- nomUsageUniteLegale: string (nullable = true)
 |-- denominationUniteLegale: string (nullable = true)
 |-- categorieJuridiqueUniteLegale: string (nullable = true)
 |-- activitePrincipaleUniteLegale: string (nullable = true)
 |-- nomenclatureActivitePrincipaleUniteLegale: string (nullable = true)
 |-- nicSiegeUniteLegale: string (nullable = true)
 |-- economieSocialeSolidaireUniteLegale: string (nullable = true)
 |-- caractereEmployeurUniteLegale: string (nullable = true)
```

A SIRET can be constructed via the following concatenation : SIREN + NIC

Your Goal

We want you to :

- Use Spark to parse and clean those Data if necessary
- Use Spark to merge the two files in a representation that you think suits the most your understanding of the Data.
- Write them to a Data Lake (For the Lake you can use HDFS locally or create a free AWS/GCP or Azure account, you are free to choose).
- Automate the pipeline execution (take care of the periodicity aspects)

Complementary Information

The idea is to spend the time you want to spend there and if you think it will take too long, below are the minimum requirements that we expect:

- A minimum code production for data read/write/merge to the Lake
- The workflow you imagined for the pipeline automation and the architecture
- Please use Python