[1. DESCRIPTION 2](#_Toc22570972)

[2. FILE STRUCTURE 2](#_Toc22570973)

[3. SCRAPER STRUCTURE 2](#_Toc22570974)

[3.1 Bank\_Name.py 3](#_Toc22570975)

[3.2 DataUtils.py 3](#_Toc22570976)

[3.3 fileUtils.py 3](#_Toc22570977)

[3.4 link\_tracker.py 3](#_Toc22570978)

[3.5 TopCompare loan Scraper.py 4](#_Toc22570979)

# DESCRIPTION

**TopCompare Scraping Tool** is a data extraction, monitoring and notification tool written in python designed for TopCompare internal use to monitor the interest’s rates of bank partners.

The scaper extracts rates data from bank providers web sites, keep track of changes over time and send notification of changes to a mail list. The extracted rates are stored as csv file in a well-defined file structure.

# FILE STRUCTURE

All the csv files are stored in a bank folder, this folder is structured as shown on the graph

File structure

**History folder**: contains all the **different** scraped rates file over time

**Update folder**: contains all the updated bank rates as csv

The current rates are in the root of the bank folder

**Principle**:

In case of recorded change:

* the current rates file is moved to the history folder,
* a file containing the updated line is generated into the Update folder
* and a new file is created with the daily extracted data as current rates.

# SCRAPER STRUCTURE

to be able to achieve this task the script is divided into 4 different python files:

* *Bank\_Name.py*
* *DataUtils.py*
* *fileUtils.py*
* *link\_tracker.py*
* *TopCompare loan Scraper.py*

## Bank\_Name.py

The file contains all the data extracting procedure for each bank, the main procedures here are:

* A procedure to request for data on the website (usually named ***requesDataFor*** or ***makeRequestFor)***. Depending on the data source which might be a pdf-file, a script or json file through a simulator, this procedure will request the website and return a data set.
* A procedure to format the data (usually named ***bankData***), the data need to be formatted with the appropriate header and type in order to exportable as csv file, this procedure will handle the formatting group them if necessary and return an exportable data-matrix.

**NB***: this procedure differs from one bank to another mainly because of the difference in dataset from each provider*

* A procedure to realize the scraping (named with the suffix ***Scraper***), using others generic procedures defined in other file this procedure will perform the data monitoring and file rearrangement.

## DataUtils.py

All the generic method used to format data are defined in this file, the procedures are used across each bank scraper If needed. The main method here are used for grouping the data and adding some attributes to the data in other to make them easy to handle or facilitate the export procedure.

## fileUtils.py

this file defines the generic methods for handling file manipulation and notification sending. It performs

* file verification and creation through procedures like ***checkToCreate(), computeFileName() or createUpdate().***
* File and frame comparison
* Frame import and export
* Update verifications and actions with the ***UpToDate ()*** procedure
* Notification procedure with the ***send\_mail\_to()*** procedure
* And some useful procedures for debugging e.g.

***no\_double***() : a functional expression that returns list of element without duplicates

***show\_double():*** shows all the duplicates in a list of elements in parameter

## link\_tracker.py

The scraper also implements a script for checking link status on TopCompare website, it makes use of request to query links on the web site and record response status.This also generates messages in case of inappropriate response.

## TopCompare loan Scraper.py

This script realizes the scraping for all banks and builds the message for notifications, it executes the Scrape procedure of all bank and stores the potential messages into a variable to be joined to notification mail.

This file implements one procedure ***tcLoanScrape().***