INTERNSHIP REPORT

Extracting and Scraping the Arrival Time for determination of NTES (National Train Enquiry System) for data used for public consumption.

Name of the Intern: Mr. Aditya Kaushal

Name of the Mentor: Mr. Narendra Ashar (General Manager Digital Railway Mobility as Service, Hitachi India Pvt Ltd.)

Internship Duration: 20th March, 2019 to 20th April, 2019

Hitachi India Pvt Ltd. | 560, Unit No. S-704, 7th Floor, World Trade Centre,
Brigade Gateway Campus, No.26/1, Dr. Rajkumar Road, Malleswaram, 055,
Rajaji Nagar, Bengaluru, Karnataka 560055.

Table of Contents

Table of Figures:

Acknowledgement

This report has been prepared for the Internship that has been done in Hitachi India Pvt Ltd, Bangalore in order to study the practical aspect of the course and implementation of the theory in the real field with the purpose of fulfilling the requirements of the course of Bachelors of Engineering Computer Science.

The aim of the Internship is to be familiar to the practical aspect and the uses of theoretical knowledge and clarifying the career goals, so I have successfully completed the Internship and compiled this report as the summary and the conclusion that have drawn from this Internship experience.

I would like to express my sincere gratitude towards my mentor **Mr Narendra Ashar (General Manager Digital Railway Mobility as a Service)** for providing this opportunity to work in Hitachi India Pvt Ltd and help me to clutch opportunities to learn the real world situation. I am also grateful to the members of Hitachi India Pvt Ltd for sharing their professional experience and helping me in every possible way. Thus, the time in Hitachi India Pvt Ltd is very audacious and supportive to my career through which I have gained valuable work experience.

I am also very thankful to Chandigarh University for letting me pursue this Internship Opportunity in Hitachi India Pvt Ltd and given me a chance to learn something new.

Mentor:
Mr Narendra Ashar (General Manager Digital Railway Mobility as a Service)

Signature:

____

Date:

Disclaimer*: Contents of this document are created by Aditya Kaushal. Did not used any Data or any other information from Hitachi India Pvt Ltd. It is exclusively based on the data available from publicly visible website of Indian Railways which are published for passengers use and communication.*

Objective:

In order to improve the Customer/Passengers experience and commitment for Indian Railways to the passengers and all the other stakeholder, it is required to have an accurate estimation of Time of Arrival of a given Coaching train at a given destination. The objective of this is to collect data to assess accuracy of the time of Arrival of Trains and a system for process improvement and corrective actions for Railways Operations.

Present Application/Solution with Indian Railways:

There are different solutions deployed by Indian Railways to make this happen, this application are technical i.e., dealing with the Operations of the Train movement and control like the COA (Control Office Application) and the National Train Enquiry (NTES). While NTES is the primary application that interact with the passengers, the COA is the non-passengers facing the system that is used for controlling the train movements. The present study done pertains to NTES since data is publicly available.

Data Collection Scheme:

Data is collected by periodically pinging the webpage(URL) in every 10 minutes interval through Scrapy as a tool which is driven by Python programming language and then exported to a CSV File Format for further manipulation and verification. The Webpage URL contains the live status of the running train in a table format which consists of ***[Station Name], [Scheduled/Act Arrival], [Scheduled/Arrival Departure]*** *as fields.* The Project only collects the last Station Name from which the train departed, the arrival time of all the stations and the sampling time from the webpage. The fields are then exported to a CSV File Format in the desired Format as ***[Local Time Stamp], [Current Station], [Arrival Time],[Sampling Time].***

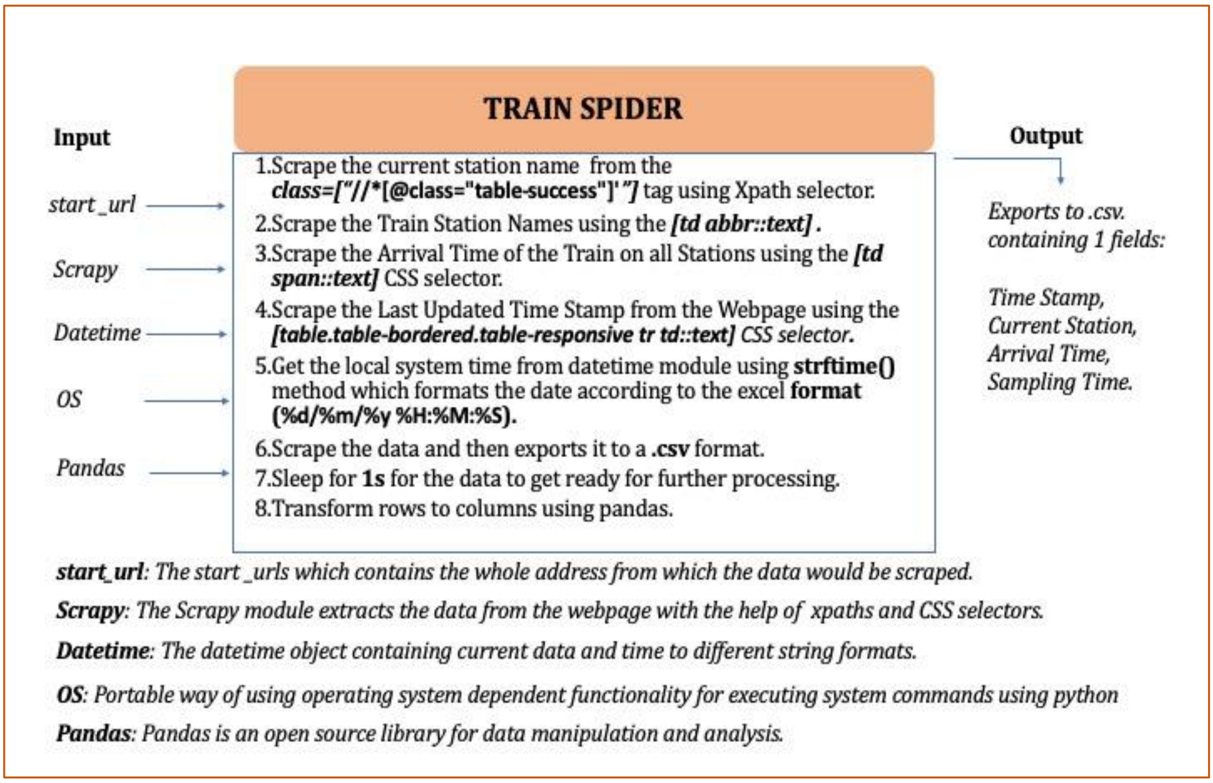
National Train Enquiry System:

The National Train Enquiry System is an integral part of Integrated Coaching Management System developed and maintained by Indian Railways. Although Indian Railways make all out efforts to run all passengers carrying trains as per their schedules an maintain their punctuality, at times for reason beyond the control of Indian Railways trains get delayed, are rescheduled from their starting stations, cancelled or diverted to another route resulting in change in the actual Arrival/Departure time from their scheduled time. To save the inconvenience caused to Rail users due to the changes in train running. National Train Enquiry System (NTES) provides information to public about expected Arrival/Departure of trains at each stopping stations, train schedule information, information about cancelled trains, diverted trains and at platforms berthing information at major stations. The main goal and objective behind NTES system to provide timely and reliable information to general public through user-friendly interfaces and PAN India accessibility has been achieved to a large extent and now the information is conveniently and reliably available to public all over the country through various delivery channels:

- Through Web Browsing
- Through Mobile Phone or
- Landline (Voice and SMS) and also
- Person through face to face enquiry and displays at all IR stations.

Module: Scraping Arrival time and Current Stations

Scrapy Train Spider scrapes the current stations and the arrival time of all the station of a specific train. The spider requires the list start_url or the URL containing the live status of the running train in a table format consisting of ***[Station Name], [Scheduled/Act Arrival], [Scheduled/Arrival Departure]*** *as fields*. This Scrapy spider scrapes desired fields and exports to the CSV File Format for further data analysis and manipulation.



1.Scrape the current station name from the *class=["//*[@class="table-success"]' "]* tag using Xpath selector.

2.Scrape the Train Station Names using the ***[td abbr::text]*** **.**

3.Scrape the Arrival Time of the Train on all Stations using the ***[td span::text]*** CSS selector.

4.Scrape the Last Updated Time Stamp from the Webpage using the ***[table.table-bordered.table-responsive tr td::text]*** *CSS selector.*

5.Get the local system time from datetime module using **strftime()** method which formats the date according to the excel **format (%d/%m/%y %H:%M:%S).**

6.Scrape the data and then exports it to a **.csv** format.

7.Sleep for **1s** for the data to get ready for further processing.

8.Transform rows to columns using pandas.

start_url*: The start _urls which contains the whole address from which the data would be scraped.*

Scrapy*: The Scrapy module extracts the data from the webpage with the help of xpaths and CSS selectors.*

Datetime*: The datetime object containing current data and time to different string formats.*

OS*: Portable way of using operating system dependent functionality for executing system commands using python*

Pandas*: Pandas is an open source library for data manipulation and analysis.*

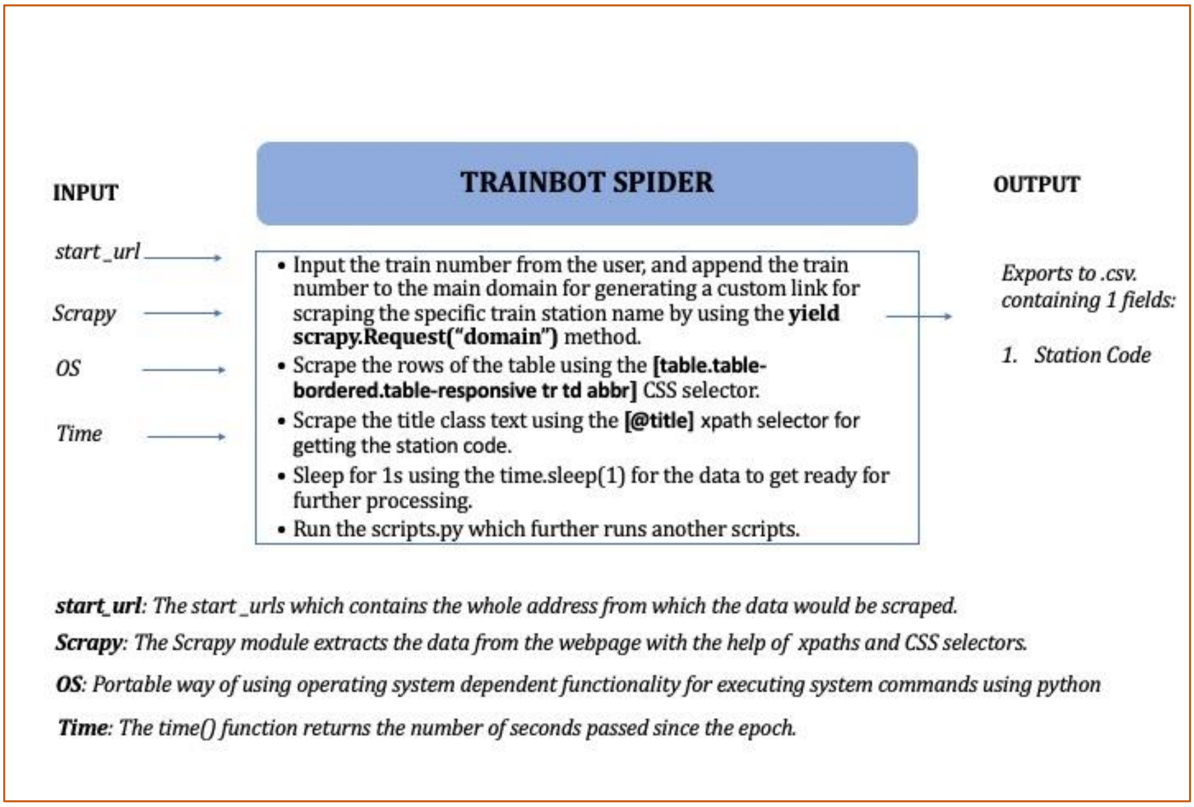
Figure 1: Scraping the [Current Station],[Station Names], [Sampling Time], [Time Stamp].

Description:

The Train Spider is responsible for scraping the Arrival Time, Current Stations (i.e., through which Station the Train Coach is passing at that instance) and the Sampling Time (i.e., the time at which the webpage was updated at the latest).The spider requires the start_url (which is a list of URL on which the spider would crawl and scrape the data for converting it to a csv formatted file). The data is scraped from the HTML tags using the XPATH and CSS selectors which are found using the developer tools in Chrome Browser. The XPATH and the CSS selectors used are mentioned in the figure above. The above spider would scrape the information i.e. the Arrival time and the Current Station in a period of every 10 minutes until the Train reaches its last Station. This would create a custom dataset for investigating and reaching a certain conclusion for verification and authenticity of data. The data exported to the CSV File Format is in a dictionary format, which is converted to a custom format though specific formatting functions and techniques. The desired format for the data is acquired after passing the data through many scripts having the usage of Pandas, CSV, XLWT and XLRD python packages.

Module: Scraping the Station Codes from a custom generated URL

The Trainbot Spider is responsible for scraping the Station Codes from the custom link generated by appending the Train Number (taken as an input) to the domain which will generate the full URL of the specific train containing the live status of the current train movement in the form of HTML Table Tags. The spider then takes the list start_url as an argument and starts scraping in accordance to the mentioned tags passed the XPATH and the CSS Selectors in the Trainbot spider.



- Input the train number from the user, and append the train number to the main domain for generating a custom link for scraping the specific train station name by using the **yield scrapy.Request("domain")** method.
- Scrape the rows of the table using the **[table.table-bordered.table-responsive tr td abbr]** CSS selector.
- Scrape the title class text using the **[@title]** xpath selector for getting the station code.
- Sleep for 1s using the time.sleep(1) for the data to get ready for further processing.
- Run the scripts.py which further runs another scripts.

Exports to .csv. containing 1 fields:

1. *Station Code*

***start_url**: The start_urls which contains the whole address from which the data would be scraped.*

***Scrapy**: The Scrapy module extracts the data from the webpage with the help of xpaths and CSS selectors.*

***OS**: Portable way of using operating system dependent functionality for executing system commands using python*

***Time**: The time() function returns the number of seconds passed since the epoch.*

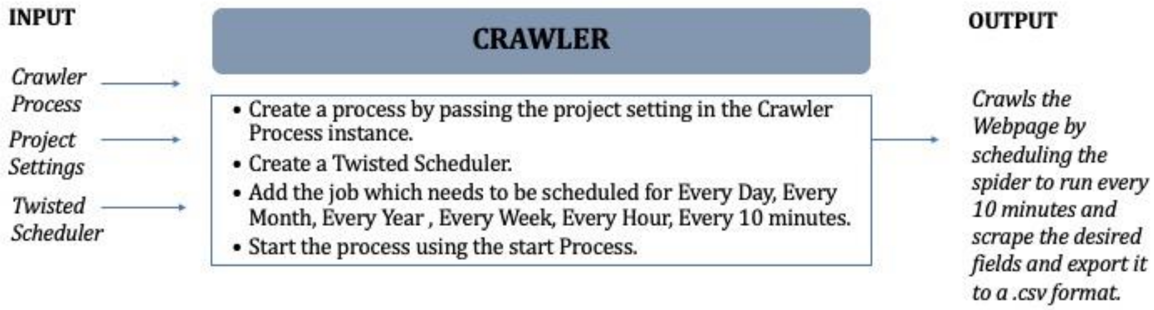
Figure 2: Scraping the Station Codes of every Station and export it to a csv format with Time Stamp and Current Station Headers.

Description:

The Trainbot spider scrapes all the Station Codes of the current route of the Train. The data is scraped using the **XPATH and CSS Selectors** which are mentioned in the above Figure. The scraped data from the Selectors is then stored in variables which is then iterated at every ith value for getting the next Station Code by using a for loop. The Station Code are then further exported to a CSV File and then converted to a Matrix using Pandas and Data frames. The Matrix is then required for further Data analysis and Manipulation. The Station Codes would also be used as the Headers for the csv file having the Time Stamp, Current Station, Arrival Time and the Sampling Time scraped from the webpage.

Module Crawler: Scheduling Train Spider to execute periodically

The Crawler is used to create a time based job for the executing the spider periodically in an interval of 10 minutes. The Crawler is utilized as script to run a spider without using the '***Scrapy crawl*** command. The Crawler utilizes special packages and modules like **Twisted Scheduler**s and **Crawlers** process for running the spider in a given interval.



- Create a process by passing the project setting in the Crawler Process instance.
- Create a Twisted Scheduler.
- Add the job which needs to be scheduled for Every Day, Every Month, Every Year , Every Week, Every Hour, Every 10 minutes.
- Start the process using the start Process.

Crawls the Webpage by scheduling the spider to run every 10 minutes and scrape the desired fields and export it to a .csv format.

***Crawler Process**: A class to run multiple Scrapy crawlers in a process simultaneously.*

***Project Settings**: The project settings gets the Settings instance with your project settings.*

***Twisted Scheduler**: Twisted Scheduler was meant to be used in Twisted applications. By default it uses the reactors thread pool to execute jobs.*

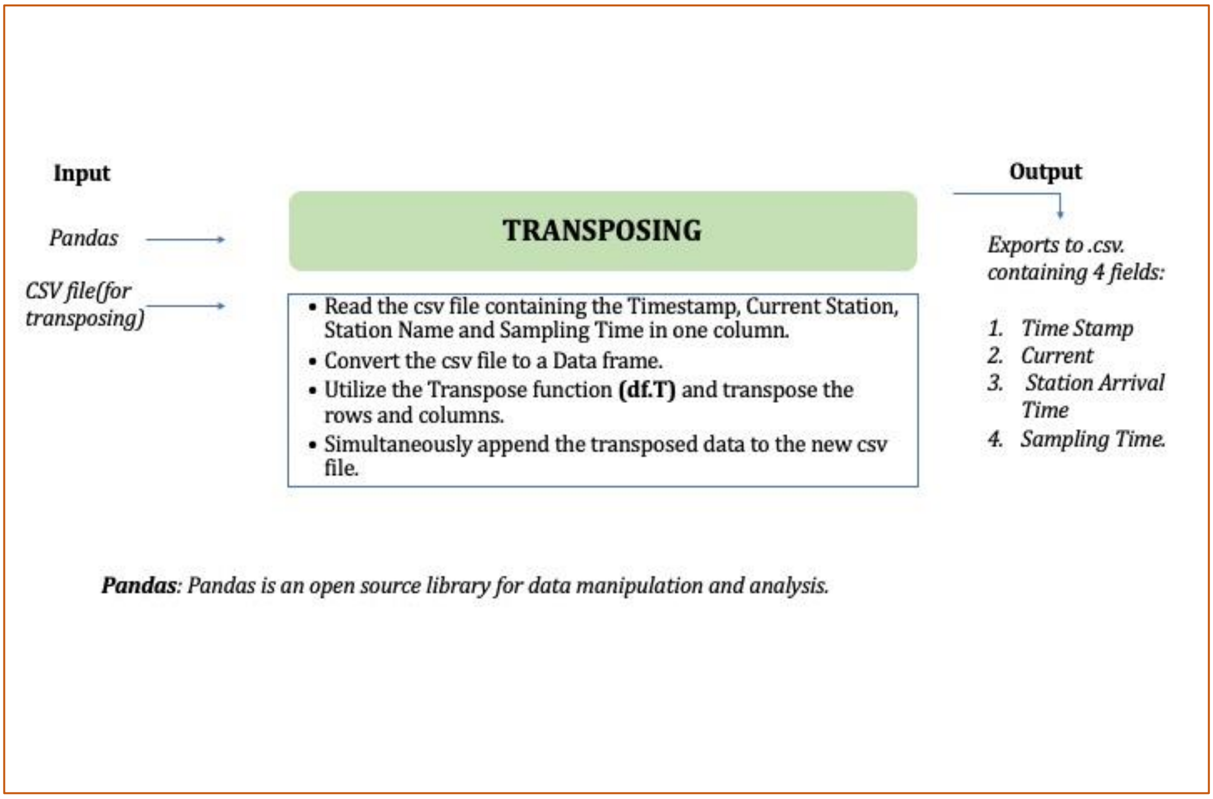
Figure 3: Schedules the script for running a Cron Job.

Description:

The crawler is like an API which is used to run the Scrapy from a script, instead of the typical way of running Scrapy via **Scrapy crawl**. Scrapy is built on the top of Twisted synchronous networking library, so it is running inside the Twisted reactor. The **scrapy.crawler.CrawlerProcess** is used to run the spider and the Twisted Reactor. The Project settings are also imported from the settings python script and loaded as an instance in the crawler script for getting the spider name and passing it as an argument in the **sch.add_job()** method for scheduling a specific spider. The crawler is then schedule the Script to run every 10 minutes and create a new row containing the *[Time Stamp], [Current Station], [Arrival Time at the Station…….n times], [*Sampling *Time].* The spider is executing periodically as a time based job in every 10 minutes interval until the train reaches the last station and then exported to a csv file for further data verification, analysis and manipulation.

Module: Transpose (Columns to Rows) => [NxN]’

The Train Spider is responsible for scraping the *[Time Stamp], [Current Station], [Arrival Time at the Station…….n times],* [Sampling *Time] in one column appended with the current date.* This data is then transposed i.e., the rows converted into columns or vice-versa for getting the desired format for data analysis.



Input

Pandas

CSV file(for transposing)

TRANSPOSING

- Read the csv file containing the Timestamp, Current Station, Station Name and Sampling Time in one column.
- Convert the csv file to a Data frame.
- Utilize the Transpose function **(df.T)** and transpose the rows and columns.
- Simultaneously append the transposed data to the new csv file.

Output

Exports to .csv. containing 4 fields:

1. *Time Stamp*
2. *Current*
3. *Station Arrival Time*
4. *Sampling Time.*

***Pandas**: Pandas is an open source library for data manipulation and analysis.*

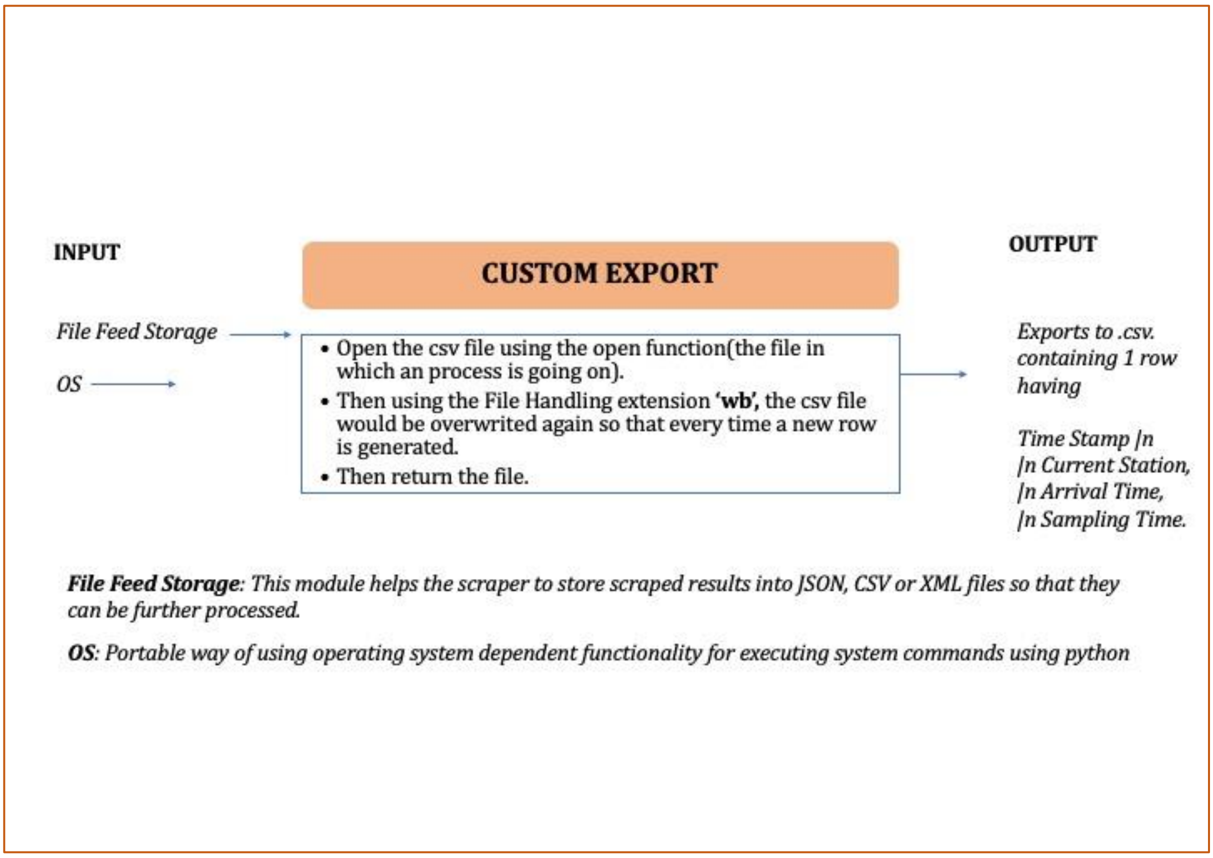
Figure 4: Transposing the columns to rows as a unit entry in csv file.

Description:

The csv which is generated by the Train Spider was in the format of single columns i.e., the *[Time Stamp], [Current Station], [Arrival Time at the Station……up to n stations],* [Sampling *Time]* in one column in the format of one below the other which is the default structure in Scrapy which is used when the data is exported it to the CSV, JSON or the XML formats. For fetching the data in the desired format it has to be transposed (i.e., the columns to rows and vice versa). Once the data has been transposed then every value in the column would be converted to a induvial row. This would then generate a custom dataset.

Module: Overwriting the csv file

The csv file from which the data would be extracted has to be overwrited because the data needs to be refreshed and be in a certain format for transposing and writing into a different csv file in the desired format. The overwriting would be done using the File Feed Storage and the operating system module required.



- Open the csv file using the open function(the file in which an process is going on).
- Then using the File Handling extension **'wb',** the csv file would be overwrited again so that every time a new row is generated.
- Then return the file.

Exports to .csv. containing 1 row having

Time Stamp /n
/n Current Station,
/n Arrival Time,
/n Sampling Time.

***File Feed Storage**: This module helps the scraper to store scraped results into JSON, CSV or XML files so that they can be further processed.*

***OS**: Portable way of using operating system dependent functionality for executing system commands using python*

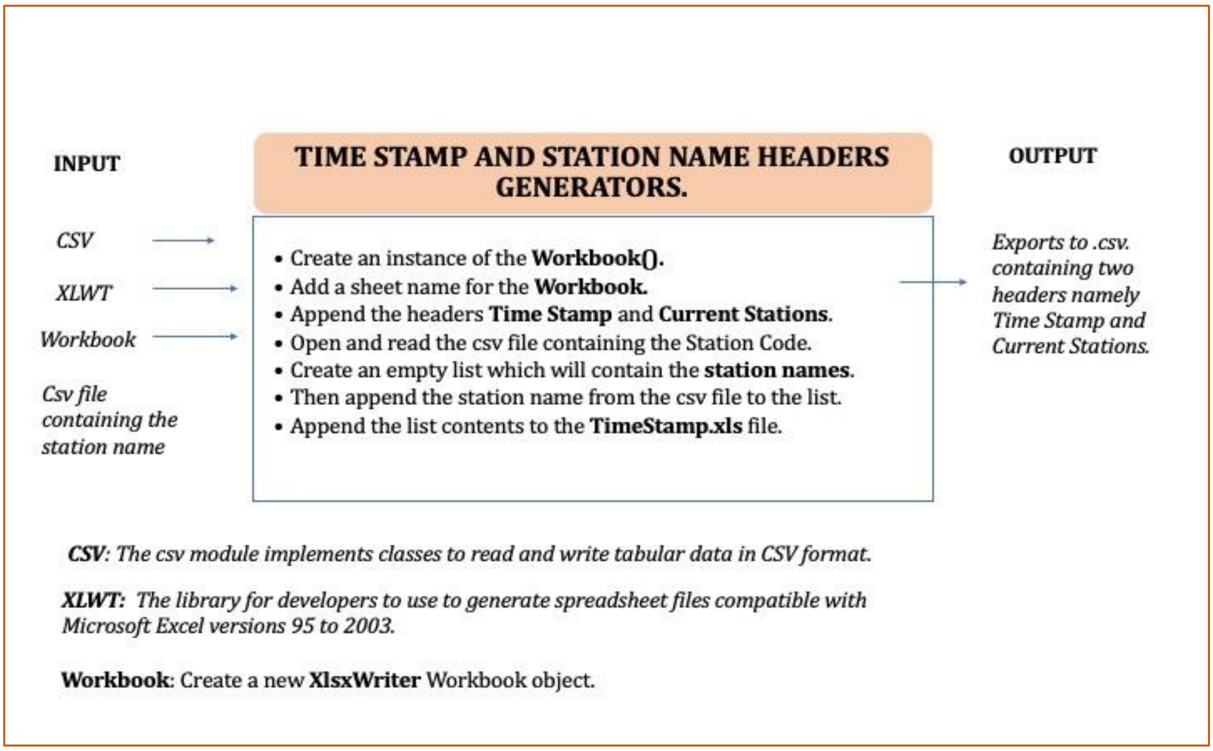
Figure 5: Overwriting the csv file through custom export.

Description:

The Custom Export would be used to overwrite the existing csv file every time the new data would be scraped from the website. The overwriting of data is necessary to remove the old data and replace it with the new refreshed data and append it in the new csv file after transposing the columns into the rows. This would be a continuous process until the train reaches the last station. The custom export file is implemented by the File Feed Storage module and the operating system(os) module for overwriting.

Module: Creating the Time Stamp and Station Code Headers

The csv file used for scraping the station codes would be used for generating the Time and Station Code headers and then copied to the different directory containing the csv file having the arrival time.


INPUT
CSV
XLWT
Workbook
Csv file containing the station name
TIME STAMP AND STATION NAME HEADERS GENERATORS.
OUTPUT
Exports to .csv. containing two headers namely Time Stamp and Current Stations.

- Create an instance of the **Workbook().**
- Add a sheet name for the **Workbook.**
- Append the headers **Time Stamp** and **Current Stations**.
- Open and read the csv file containing the Station Code.
- Create an empty list which will contain the **station names**.
- Then append the station name from the csv file to the list.
- Append the list contents to the **TimeStamp.xls** file.

CSV: *The csv module implements classes to read and write tabular data in CSV format.*

XLWT: *The library for developers to use to generate spreadsheet files compatible with Microsoft Excel versions 95 to 2003.*

Workbook: Create a new **XlsxWriter** Workbook object.

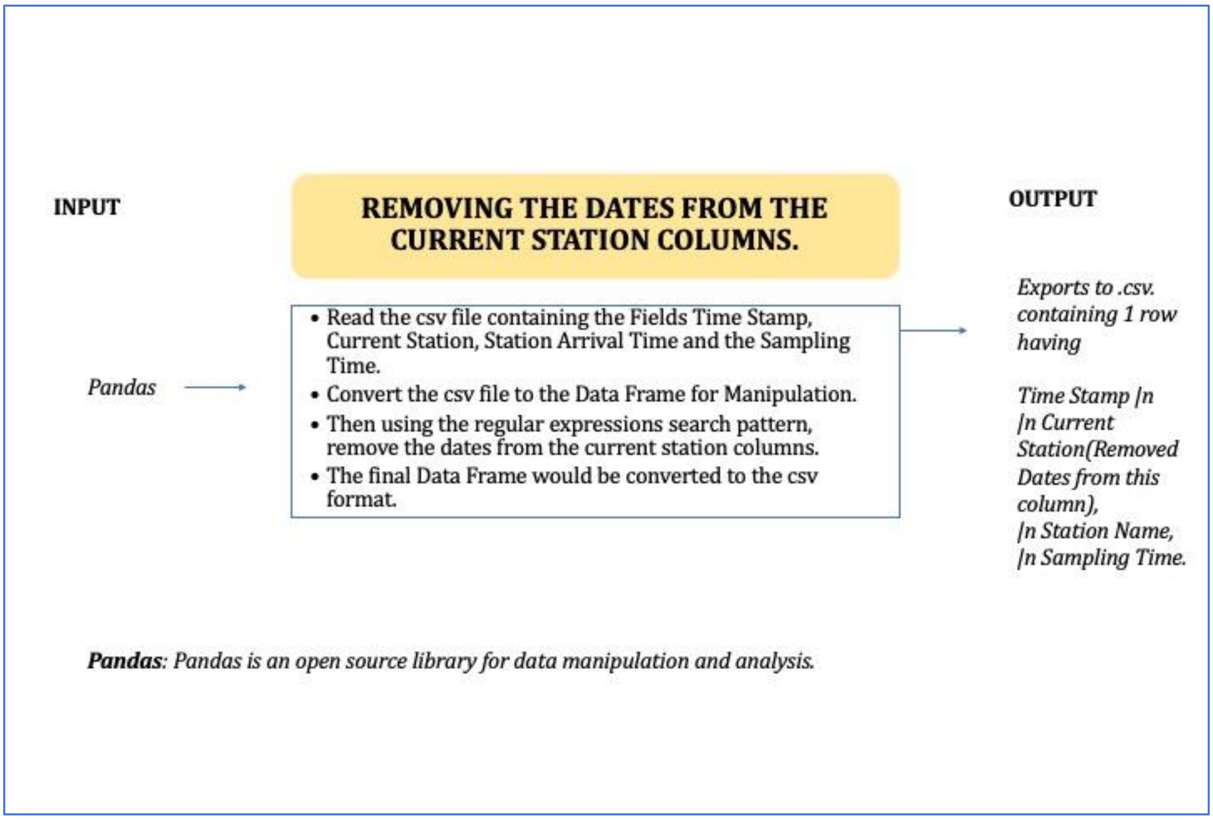
Figure 6: Generating the Time Stamp and Current Station headers

Description:

The Time Stamp and Station Name Codes are two of the headers of the csv file containing the arrival time and the local system timestamp of the trains. The station name codes are taken from the matrix csv file generated. An empty list is generated then after iterating the matrix csv file headers, the station names are appended to the empty list. Furthermore, the Time Stamp and Current headers are appended to the list and then exported to the csv file.

Module: Remove Dates appended to Current Station Column

The local system date is appended to all the items of the list which are scraped from the webpage like the arrival time and the sampling time. The date is appended to the current station column also which needs to be removed for getting the desired format for further data verification and manipulation.



INPUT

REMOVING THE DATES FROM THE CURRENT STATION COLUMNS.

OUTPUT

Pandas →

- Read the csv file containing the Fields Time Stamp, Current Station, Station Arrival Time and the Sampling Time.
- Convert the csv file to the Data Frame for Manipulation.
- Then using the regular expressions search pattern, remove the dates from the current station columns.
- The final Data Frame would be converted to the csv format.

→

Exports to .csv. containing 1 row having

Time Stamp /n
/n Current Station(Removed Dates from this column),
/n Station Name,
/n Sampling Time.

***Pandas**: Pandas is an open source library for data manipulation and analysis.*

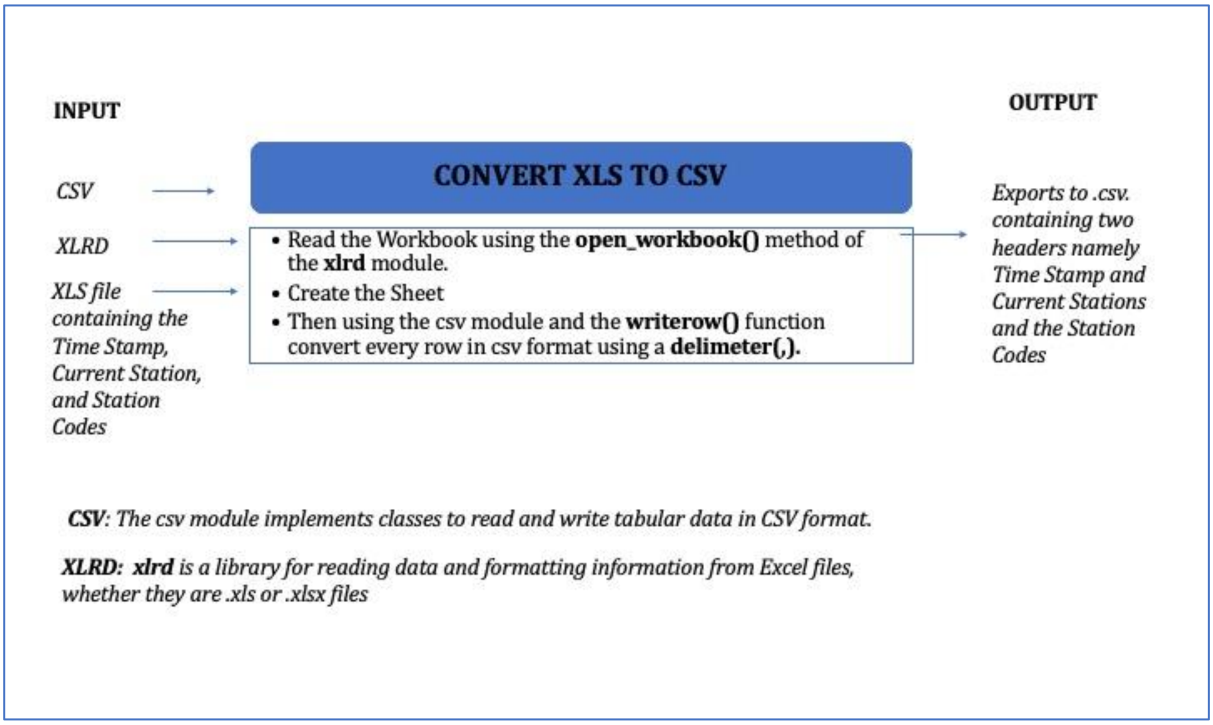
Figure 7: Removing the Date from the Current Station Column

Description:

The current station columns should contain all the stations through which the train would be passing through. The dates are appended to all the fields as well as the current station. The pandas are helpful in manipulating the current station column for removing the dates. The regular expression search has been useful for removing a similar pattern that the dates follow. The csv file has been converted to the Data frame and then the dates are removed from the Current Station Column.

Module: Converting XLS file to CSV File.

This module is used to convert the xls file to csv file. The



INPUT

CSV

XLRD

XLS file containing the Time Stamp, Current Station, and Station Codes

CONVERT XLS TO CSV

- Read the Workbook using the **open_workbook()** method of the **xlrd** module.
- Create the Sheet
- Then using the csv module and the **writerow()** function convert every row in csv format using a **delimeter(,).**

OUTPUT

Exports to .csv. containing two headers namely Time Stamp and Current Stations and the Station Codes

***CSV**: The csv module implements classes to read and write tabular data in CSV format.*

***XLRD: xlrd** is a library for reading data and formatting information from Excel files, whether they are .xls or .xlsx files*

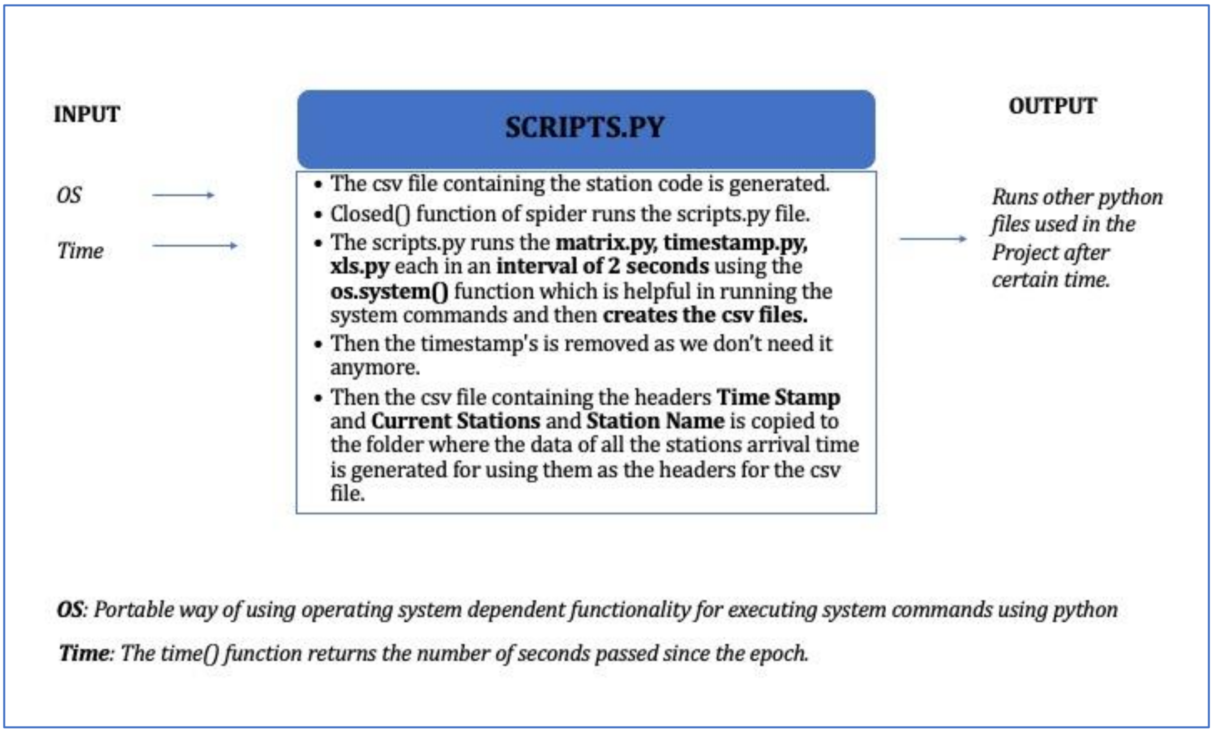
Figure 8: Convert the XLS To CSV

Description:

This module is used for converting the TimeStamp.xls to convert it to the TimeStamp.csv for manipulating the csv file further. The TimeStamp.xls is the excel sheet which needs to be converted using the xlrd and csv python module packages. The packages converts the XLS file taken as an input and converting it into a csv(comma separated values).

Module Scripts

The scripts.py module is responsible for executing different python modules using the os package. The OS package is a portable way of running system commands using function named as os.system(‘ command’). This script is used for consolidating the whole project into one script. The scripts.py consists of three commands which are used to run the timestamp.py and matrix.py.



INPUT

OS

Time

SCRIPTS.PY

- The csv file containing the station code is generated.
- Closed() function of spider runs the scripts.py file.
- The scripts.py runs the **matrix.py, timestamp.py, xls.py** each in an **interval of 2 seconds** using the **os.system()** function which is helpful in running the system commands and then **creates the csv files.**
- Then the timestamp's is removed as we don't need it anymore.
- Then the csv file containing the headers **Time Stamp** and **Current Stations** and **Station Name** is copied to the folder where the data of all the stations arrival time is generated for using them as the headers for the csv file.

OUTPUT

Runs other python files used in the Project after certain time.

***OS**: Portable way of using operating system dependent functionality for executing system commands using python*

***Time**: The time() function returns the number of seconds passed since the epoch.*

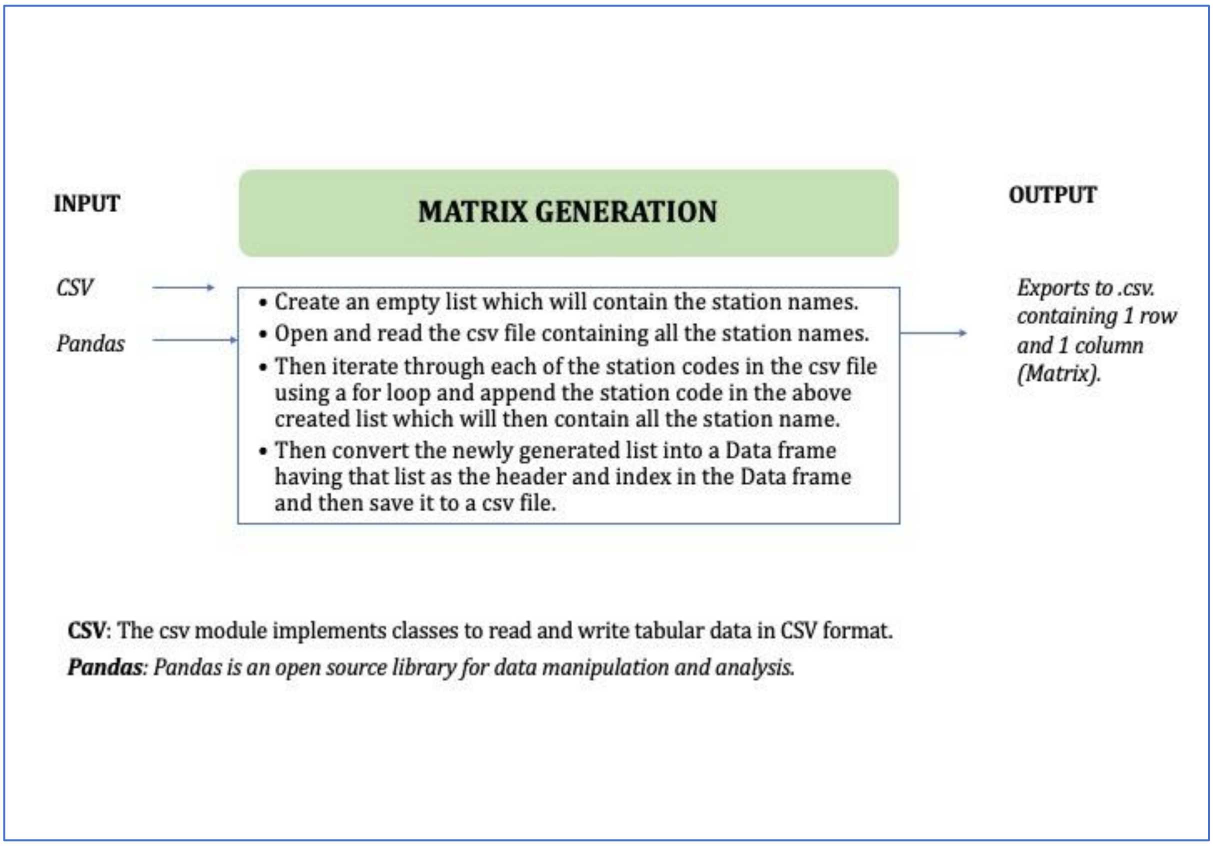
Figure 9: Consolidating the Project Using the Scripts.py

Description:

The scripts.py is a python script which is using the os and the time module for running basic python commands for executing three different python files one after the other. The scripts.py is useful for consolidating the whole project into one command. As, soon as the Trainbot Spider scrapes the data from the webpage (URL), the closed() function of the spider is called, which contains the **os.system(‘scripts.py’).** This command is responsible for running the scripts.py command which automatically generates the matrix, the headers files(Current Station and Time Stamp), without calling them explicitly.

Module: Matrix Generation

Matrix is a linear data structure consisting of the station codes of the current train route as the headers and the indexes in the csv file format.



- Create an empty list which will contain the station names.
- Open and read the csv file containing all the station names.
- Then iterate through each of the station codes in the csv file using a for loop and append the station code in the above created list which will then contain all the station name.
- Then convert the newly generated list into a Data frame having that list as the header and index in the Data frame and then save it to a csv file.

Exports to .csv. containing 1 row and 1 column (Matrix).

CSV: The csv module implements classes to read and write tabular data in CSV format.

Pandas*: Pandas is an open source library for data manipulation and analysis.*

Figure 10: Matrix Generation for Data Verification and Analysis

Description:

The matrix is generated with the help of pandas and csv python module packages. Pandas converts the CSV file into Data Frame for further Data Processing and verification of the Arrival Time of the Trains at different train stations. The Analysis done on the Matrix Generation is beyond the scope of this one month project.

Findings

After extracting the Arrival Time of the Important Stations, the Arrival Time of the intermediate stations other than the ones listed in NTES schedule are also being extracted up to the Station the Train has reached. But due to the operational conflicts or the Normal Operations of the current Railway Operations, the Arrival time of the Intermediate Station/Non-Stopping Station are being flushed out after train departs from that Station.

Conclusion and Scope for further Analysis

The Data collected is the Arrival Time of all the Stations of Train (i.e., Important Stations and Intermediate Station), with the Local Time Stamp, Current Station and the Sampling Time. Furthermore, the data collected can be further used for statistical analysis and verification of the accuracy of the Arrival Time.