**DataMigrator API**

**User manual**

**DataMigrator is an agile and effective Python based, user friendly opensource migration tool that can be used to build ‘data migration pipeline’. It essentially connects and understands a set of sources and syncs and helps quickly build a smooth migration pipeline with small number of iterations.**

* Currently, this tool supports 5 Data Source types which means user can migrate data from all these 5 different source types to any of the 5 types he/she wants.
* This tool won’t be needing much coding experience from its user.
* This tool can transform in stand-alone or distributed mode. We have SPARK-ified our tool which gives it ability to migrate data from big data ecosystem also in a distributed way.
* Following are the Data Source type supported by **DataMigrator**:

1. ***MS-EXCEL*** (keyword used in configuration file is ‘**xls’** )
2. ***PARQUET*** (keyword used in configuration file is ‘**parquet’** )
3. ***CSV*** (keyword used in configuration file is ‘**csv’** )
4. ***JSON*** (keyword used in configuration file is ‘**json’** )
5. ***PICKLE*** (keyword used in configuration file is ‘**pickle’** )

**How to use this API?**

It consists of 3 steps:

**Step 1:**

Create 3 configuration files which consist all your requirements. Following are the detailed explanation of these files:

1. **req\_conf.json :**

This file contains the details about the input/output data source type and path of configuration file for source type details. It contains following keys which user has to change according to his requirement.

{

    "input\_type":"json",

    "input\_conf\_path":".../DataMigrator/config/Input/input\_json.json",

    "output\_conf\_path":".../DataMigrator/config/Output/output\_csv.json",

    "output\_type":"csv",

    "distribute": "True"

}

**“distribute”** key is a Boolean variable which tells the **DataMigrator** whether use the spark distributed computing or not. If it is True, user has to send the running spark instance also to utilize the capabilities of spark environment and deal big data with ease.

1. **‘input\_type’.json :**

This file contains the parameters required to read the source data. For e.g. following is a sample **parquet configuration file** named ***‘input\_parquet.json’***.

{

    "path":".../DataMigrator/Data/input/input\_data.parquet",

    "engine":"fastparquet"

}

User can add as many parameters as he wants which is required for reading parquet file in the format of key-value pair of JSON structure.

1. **‘output\_type’.json :**

This file contains the parameters required to save the data in required format**.**For e.g. following is a sample of **CSV configuration file** named ***‘input\_csv.json’***.

{

    "path":".../DataMigrator/Data/output/",

    "file\_name":"output\_data.csv",

    "coalesce\_num":3

}

Here we have one more key “**coalesce\_num**”. This is required when we want to run our API in distributed environment (Big Data environment). It tells **DataMigrator** the number of partitions user wants to divide the file before writing.

**Step 2:**

User needs to create a spark instance to provide the distributed capability to the **DataMigrator**. If user wants to run the API in single mode or local mode only, he/she can ignore this step and directly jump to the 3rd step.

**Step 3:**

After creating all the configuration files and spark instance (if required), User has to import our DataMigrator API and pass the path of ‘***req\_conf.json***’. Execute this command and you are done.

**Congratulations!! You just migrated your data from one source type to other.**