International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 07 | July 2022 www.irjet.net p-ISSN: 2395-0072

Design and Developing a Snaplogic Pipeline for Automating Batch Import of Records

Rohan V K[1], Anusha L S[2]

[1]*Student, RV college of Engineering, Karnataka India*
[2]*Professor, RV College of Engineering, Karnataka India*

Abstract - *Data is imported through files or through Application programming interface (API) to database on cloud at enterprise level. When the data is present in a document it needs to be processed before it is updated to the database. The processing of document involves steps like file reading, input parsing, data validation, API call and generating final report of the records imported from the document. The above steps have to be repeated when a new document is to be imported. The Snaplogic platform provides various tools called snaps which can be integrated to process the document. The snaplogic pipeline can be triggered using datahub platform which will run the pipeline when a new document is to be imported thus, automating the process of importing records.*

Key Words: **Data validation; Datahub; Snap; Input parsing; Workflow;**

1. INTRODUCTION

Importing bulk records involves processing of records in groups. Batch process is used only if there are more then one records to be imported as it would require to make the API call every time a user updates and is a tedious task. Datahub is used to create workflow which compiles to a script. This script will trigger the snaplogic pipeline which will process the records in the file. The snaplogic pipeline is constructed with two or more sub pipelines for modularity and reuse purpose. The pipeline contains various snaps which are integrated to perform steps like reading file from the database, parsing the input records, making the API request and to generate the final report of the records imported. The pipeline created will be cost effective as is will import only the valid records and discards invalid records which saves unnecessary update to the database. The above process is automated using datahub which will trigger the pipeline to process the records when a new file needs to be imported. The Datahub workflow takes batch file as input and returns the document which contains the information of the processed batch file as output.

2. DESIGN OF WORKFLOW AND PIPELINE

The datahub workflow consist of following components:

Input batch file, start component, pipeline parameter, execute snaplogic pipeline, stop and output file

2.1. DATAHUB WORKFLOW

The workflow (fig 1) consists of various blocks which are required for the run time execution of the pipeline and the workflow. The input file which has to be processed is sent into the Datahub workflow. This input file contains raw information's which is processed by the pipeline later. The Datahub workflow consists of various components like the start component which is used to trigger the workflow, pipeline parameter which contains information where the file is stored before and after processing. Some of the other components include execute Snaplogic pipeline which is responsible for running of the pipeline. The Snaplogic pipeline consists of various stages which are executed in order to process the input file. It consists of sub pipeline file reader, data validation, API and final report generator sub pipelines. The stop component is used to stop the execution of the pipeline. The final output contains the information of the imported records.

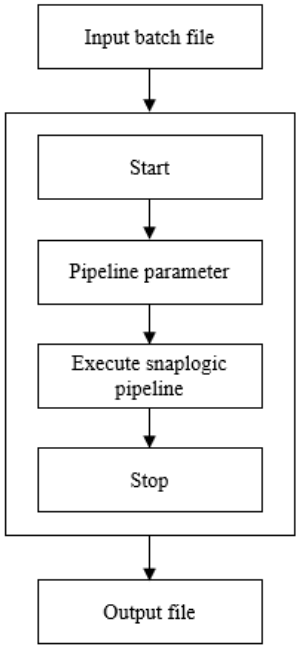

Fig.1. Workflow layout

2.2 SNAPLOGIC TOOL

Snaplogic provides cloud integration technologies that enable users to link cloud-based information and programmes with both on-premises and cloud-based enterprise software. Even business customers with less technological expertise can access and combine data from many sources because of the products' architecture.

The Snaplogic Integration Cloud includes Snaplex, Snaplogic manager, Snaplogic monitoring and Snaplogic designer. Snaplex is a self-upgrading network of execution that transmits data between programmes, databases, files, social media, and large data sources. The Snaplex can elastically scale up or down when operating in the cloud, depending on the amount of data being processed or the latency specifications of the integration flow. The Snaplex can be set up to operate locally while integrating local systems. Snaplogic Designer is a tool for building integration workflows (also known as pipelines), which are collections of Snaps connected in a specific order. There is no need for scripting because snaps may be joined using a visual drag-and-drop interface.

Snaplogic Manager is a tool for building integration workflows (also known as pipelines), which are collections of Snaps connected in a specific order. There is no need for scripting because snaps may be joined using a visual drag-and-drop interface. Snaplogic Monitoring Dashboard is an interface for monitoring the effectiveness of integration workloads locally or remotely. For complete remote visibility over planned and real-time bulk integrations, the Monitoring Dashboard is accessible via a browser and on mobile devices like the iPad.

2.3 SNAPLOGIC PIPELINE

The snaplogic pipeline (fig 2) contains following sub pipelines which are designed in modules called sub pipelines and these sub pipelines can be reused in other main pipelines.

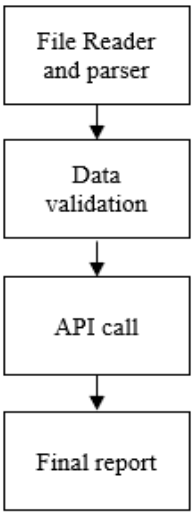

Fig.2. Snaplogic pipeline flow

The file reader stage is used to read input file which is stored in the database in the Snaplogic platform. The parser snap is used to extract the values. Data validation step is used to validate the data based on certain criteria which will filter out valid and invalid records. API call which will only consume valid records and discard invalid records. The last sub pipeline is used generate report of imported batch file.

3. IMPLEMEMTING SNAPLOGIC PIPELINE

The pipeline is implemented in by creating three sub pipelines and specification of each sub pipeline is discussed below.

3.1 FILE READER AND PARSER

The layout of the sub pipeline (fig 3) includes various snaps. The file reader stage is used to read input file which is stored in the database in the Snaplogic platform. This step is important as the pipeline would not run if the file is not imported. The file should be a text document and other files are discarded. The parser snap is used to extract the values. The values extracted are formatted in object notation which later can be used to append to object and sent as a request body.

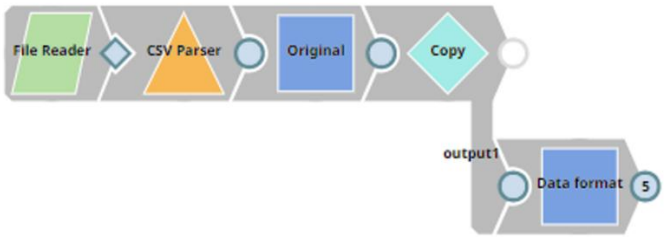

Fig.3. File reader and Parser sub pipeline.

The file reader contains label as attribute and need to specify file name which is stored in the database. Number of retries need to be set to zero. CSV parser snap is used to parser the input records. The mapper snap contains label as attribute and value is set to original. And rest of the attributes are unchecked. The mapper passes all the columns from the file reader. The data format snap contains column which will be extracted from the file and processed. The attributes extracted are segment, state. region, category, sales and profit.

3.2 DATA VALIDATION

Data validation step is used to validate the data based on certain criteria which will filter out valid and invalid records. The classification of the records is done based on the minimum length, maximum length and required fields. A record is assigned as valid only if the above conditions are satisfied. This step is used to make API call which will only consume valid records and discard invalid records. The records are then updated in the database which can be later viewed. The layout shows (fig 4) more mapper snaps are used which are represented in blue color. The mapper snaps are used to map previous stage output to current stage output. The order of the variable declared in the snap will continue in the next stage.

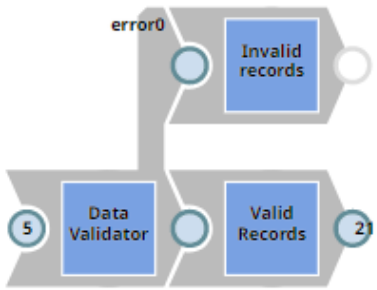

Fig.4. Data validation sub pipeline

The data validation table is created based on the table values the records will be assigned as valid or invalid. If the filed with constraint as required is set to true implies the value needs to be present. Each record is validated based on the constraints (table I).

Table -1: Data validation constraints

Source path	*Constraints*	*Constraints value*
User	MinLength	1
User	Required	true
Sales	Required	true
State	Required	true
State	MaxLength	20
Region	MaxLength	100
Region	MinLength	1
Profit	Required	true

3.3 API CALL

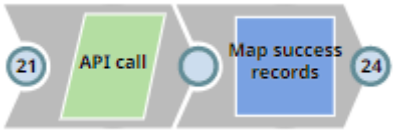

Fig. 7. API call sub pipeline

The API Snap can be used to read binary stream of data from an HTTP source as your input and write it to the HTTP Uniform Resource Identifier (URI), using the PUT or POST method. The response from the target is available from the output views in either record or binary form. The Snap can also be used to read the output of the XML Write building block as its input, and write the output to an application's REST software interface. The API snap is used to add request to each record which are set as valid. The invalid records are discarded and not allowed to pass to the API. Map success records assigns valid records to the attribute. It contains label as attribute with value map success records and passed original path along with entity message. The API is a post request it contains headers like language, content type, status and request body. The request body contains the record and the records gets updated to the database as all the validations are verified.

3.4 FINAL REPORT

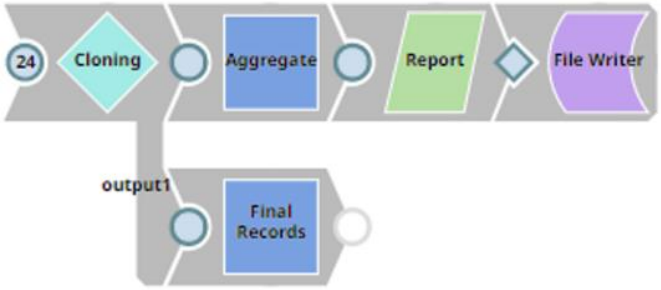

Fig. 6. Final report sub pipeline

The last step of the pipeline execution is used to generate the report of the input file. The report gives the information like total profits, total units and other information.

Aggregate snap uses the Group By support, the Snap is used to apply aggregate functions to input data. It has the ability to compute a single scalar value from a collection of inputs. The aggregate snap helps to add aggregate functions like sum, min, max and average functions. These functions can be used to append to the final report. It contains label attribute with value as aggregate. Snap to inject a block of text that may contain expression language, pipeline parameters, and binary data. Final record mapper is used to map original records to

File Writer Snap helps read a binary data stream from the input view and write it to a certain file location. Label value is set to file writer and file name is specified as final report with text document. The file action is set to overwrite which will replace the file in the database.

4. RESULTS

The results from the snaplogic contains the success message. It also contains attributes like total profit which includes sum of all profits from the users. Average profit includes average profit of the users. The total sales are sum of the category which have contributed to the sales. The total users are the count of records. The sample output file with file name as finalreport.txt is shown. (fig8)

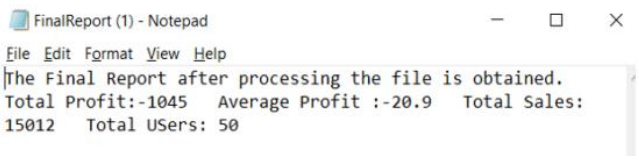

Fig. 8. Output as text file from pipeline

The comparison of snaplogic with API call shows that it would cost significantly more units to update the database as the invalid records also will be updated to the database. The use of datahub also automates the process of batch import of records. The automation will save time and resource which will enhance the application

Type	Valid record	Invalid record	Cost
API	70000	30000	100 units
Snaplogic	70000	30000	70 units

5. CONCLUSION

The Datahub workflow is designed and developed in open-source platform along with snaplogic. The snaplogic pipeline is created by interconnecting snaps. The results are then obtained as a text document. The comparison of the snaplogic and API is also evaluated. The snaplogic pipeline returns the results inform of text document which gives information on total sales, total profit and average sales. These values can be used for business purpose and automating the pipeline with Datahub also saves time. The comparison of Snaplogic

REFERENCES

[1] P. Acosta-Vargas, S. Luj´an-Mora, and L. Salvador-Ullauri, “Evaluation of the web accessibility of higher-education websites,” in 2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET), IEEE, 2016, pp. 1–6. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] F. Andrade-Chaico and L. Andrade-Arenas, “Automation of the short message service (sms) delivery in a telecommunication company with python and batch files,” in 2020 IEEE Congreso Bienal de Argentina (ARGENCON), IEEE, 2020, pp. 1–5.

[3] .M. Campoverde-Molina, S. Lujan-Mora, and L. V. Garcia, “Empirical studies on web accessibility of educational websites: A systematic literature review,” IEEE Access, vol. 8, pp. 91 676–91 700, 2020.

[4] W. A. R. W. M. Isa, M. A. Aziz, and M. R. B. A. Razak, “Evaluating the accessibility of small and medium enterprise (sme) websites in malaysia,” in 2011 International Conference on User Science and Engineering (i-USEr), IEEE, 2011, pp. 135–140.

[5] W. A. R. W. M. Isa, A. I. H. Suhaimi, N. Ariffrn, N. F. Ishak, and N. M. Ralim, “Accessibility evaluation using web content accessibility guidelines (wcag) 2.0,” in 2016 4th International Conference on User Science and Engineering (i-USEr), IEEE, 2016, pp. 1–4.

[6] B. Roth, B. Volz, and R. Hecht, “Data integration systems for scientific applications,” in OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”, Springer, 2010, pp. 110–118.

[7] X. E, J. Han, Y. Wang, and L. Liu, “Big data-as-a-service: Definition and architecture,” in 2013 15th IEEE International Conference on Communication Technology, 2013, pp. 738–742. doi: 10.1109/ICCT.2013.6820472.

[8] I. Serna-Marjanovi´c, A. Tanovic, and A. Cerimagic, “Accessibility standards and their implementation in custom data-driven maps,” in 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), 2020, pp. 1674–1679. doi: 10.23919/MIPRO48935.2020.9245417.