Nandetechai

DATA INGESTION AND ETL BOILERPLATE

Presented by: NandetechAI

OVERVIEW

01 Introduction	02 Data ingestion	03 ETL
04 Boilerplate steps	05 Boilerplate review	06 Conclusion

INTRODUCTION

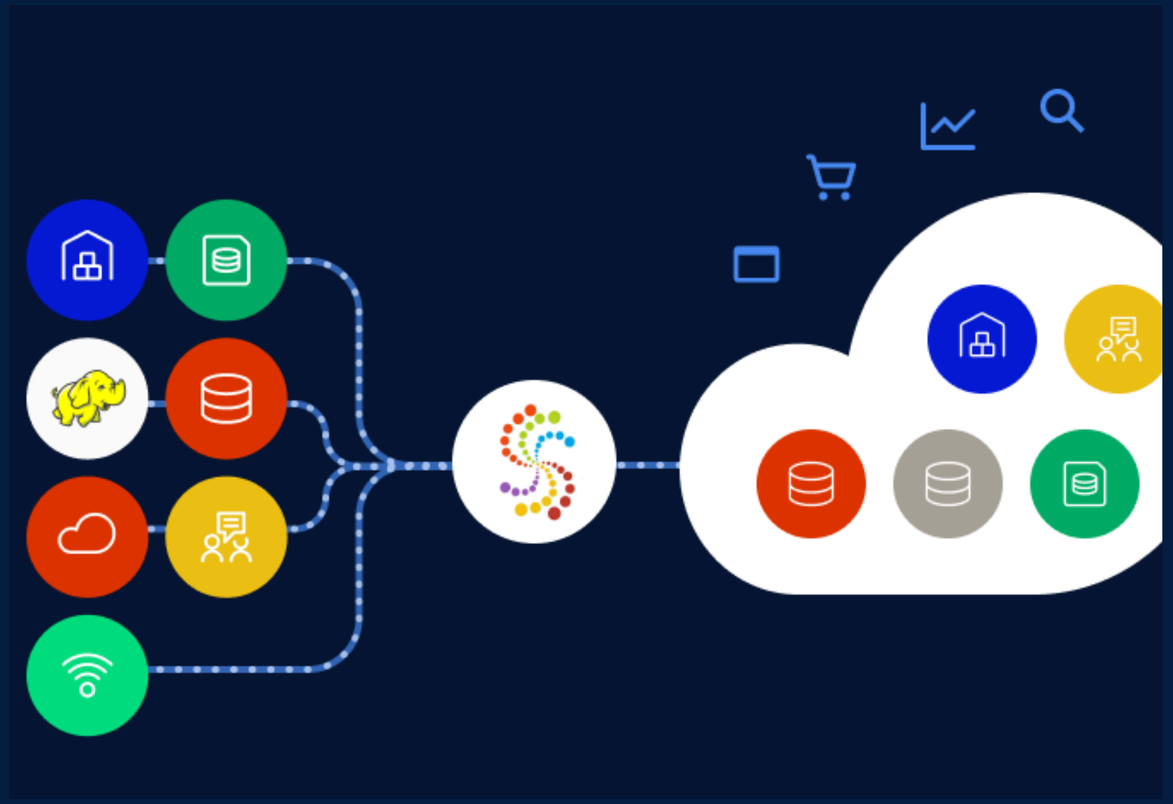
>Data ingestion is an important process in data management. The process involves collecting and importing data from different sources into a storage or processing system. This is a foundational step in data workflows, enabling subsequent data analysis, reporting, and operational activities.

>For better data practices and reliable results, it very crucial to correctly extract and process data. Understanding and effectively managing the data ingestion process can greatly enhance your ability to leverage data for strategic insights and operational efficiency.

DATA INGESTION

This is a process where large, assorted data files are imported from multiple, various sources into a single , cloud-based, storage medium, or a database where it can be accessed or analyzed when required.

This process ensures that data from disparate sources is collected, structured, and available for analysis and processing. Data ingestion process is differentiated into two types; Real-time and Batch processing



TYPES OF DATA INGESTION

BATCH PROCESSING

This process collects data in batches at regular intervals from different sources to destination. Used when data access is not emmediately required. This approach is useful for processing large volumes of data at once but can introduce latency.

REAL-TIME PROCESSING

This is a process of collecting data from multiple, various data sources in real-time. This approach is used for applications requiring immediate insights and quick decision-making.

ETL DATA INGESTION

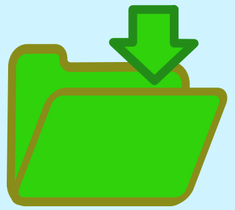
ETL is a process of combining or integrating data from multiple sources into a large, central repository called data warehouse. ETL stands for Extraction, Transformation, and Loading of data.

Extraction

Involves collecting data from different source systems. The sources can be diverse, including; database, files, streaming sources, real-time data feeds, logs, and external services.

Transformation

Involves converting the extracted data into a format suitable for analysis or storage. This step can include various operations, such as; data cleaning, data formatting, data aggregation, data enrichment, and data Integration.



Loading

Involves writing the transformed data into the target repository, such as; data warehouse, data lake, database.

BOILERPLATE

Standardized code snippets or structures that serves as a foundation for new work, and are reused across multiple projects.

STEPS FOR CREATING A BOILERPLATE

01 Setting up your development environment

This the first step, where you ensure you have your VScode and python installed. You also set up the project directory.

02 Set up your virtual environment

You set up the environment you are going to work on for your project to avoid, interrupting your other ongoing project.

STEPS FOR CREATING A BOILERPLATE

03 Create the flask application

In this step, you must install (if you have not yet) and set up the flask application.

04 Create your directory structure.

This step includes creating your required project directory (folders) and inserting the skeleton files that will be used in your boilerplate to perform the requires processes. You also insert codes in your files.

05 Testing the application

In this step, we try run the flask app, to test whether it works, and if it works like we wanted it to

STEPS FOR CREATING A BOILERPLATE

06 Documenting your boilerplate

After completion of your flask application and creating a boilerplate you have to document it and save your work, you may use applications such as the Github.

07 Updating your boilerplate

Make use of the user feedback to update your boilerplate when and where needed.

*The steps written above will be displayed on the demonstration boilerplate for more coding information.

STEPS FOR CREATING A BOILERPLATE (CODE SNIPPETS)

01 Setting up your development environment

```
# Create a new directory for your project
mkdir data_ingestion_boilerplate

# Navigate into the new directory
cd data_ingestion_boilerplate
```

#Open the terminal on your VScode and run these commands to set your working environment

02 Set up your virtual environment.

```
python -m venv venv
```

#Create you environment in the terminal

```
.\venv\Scripts\Activate.ps1
```

#Activate your environment in your PowerShell

#In case you run into an error, mostly the policy error, run the following command then proceed.

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope Process
```

STEPS FOR CREATING A BOILERPLATE (CODE SNIPPETS)

03 Create the flask application

```
pip install Flask
```

#Run the command on your terminal to install Flask (if not have done so).

04 Create your directory structure.

This step includes creating your required project directory (folders) and inserting the skeleton files that will be used in your boilerplate to perform the requires processes.

```
cd C:\path\to\data_ingestion_boilerplate
```

#Open your PowerShell. Run the command to navigate to your directory

```
mkdir templates static etl uploads
```

#Create the main directories (folders).

```
New-Item -Path app.py -ItemType File
```

#Create your app.py

STEPS FOR CREATING A BOILERPLATE (CODE SNIPPETS)

>The next step is to create your empty files for your directory. (Run these commands on PowerShell.

```
New-Item -Path templates\upload.html -ItemType File
```

```
New-Item -Path etl\__init__.py -ItemType File
```

```
New-Item -Path etl\extract.py -ItemType File
```

```
New-Item -Path etl\transform.py -ItemType File
```

```
New-Item -Path etl\load.py -ItemType File
```

>These files enable the functions of your boilerplate. You will then insert the required codes in each of the files. Your codes contain the significant functions enabling the functionality of your boilerplate. Your codes must be accommodative/ flexible to enable an ability of your boilerplate to be useful in different project. (The codes examples are included in the files uploaded in the project Github repository page.

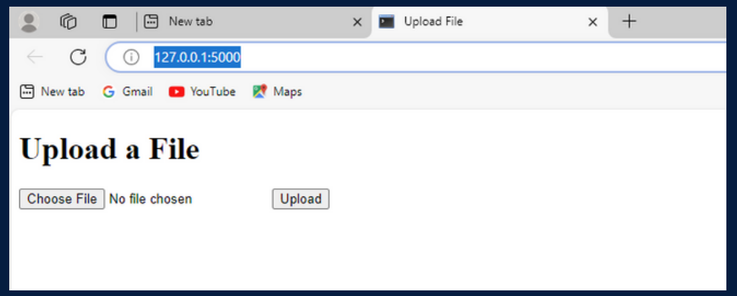
STEPS FOR CREATING A BOILERPLATE (CODE SNIPPETS)

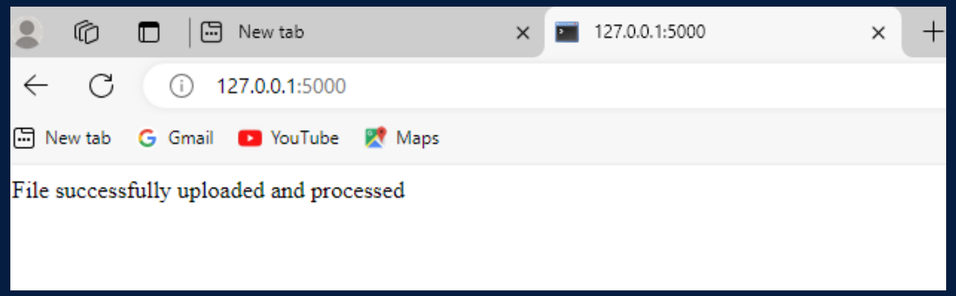
05 Testing the application

```
python app.py
```

#Run the command on you terminal/ PowerShell to run the application.

> Open our web browser and go to http://127.0.0.1:5000/ and you will see the web interface with an upload form.





> This will be displayed on your web browser. You will navigate to “Choose file” and select the file you want to upload. Navigate to “Upload” after selecting the file. If your file was successfully uploaded, extracted, transformed and loaded to path, it will display the “ File successfully uploaded and processed”. Then your boilerplate functions well. If unsuccessful it will display “No selected file or invalid file type”, you will need to navigate back to your code and ensure it ran well.

Boilerplate

Link: *(Boilerplate files available on the Github repository.)*

Conclusion

The created boilerplate allows you to load up multiple files, extract data, transform it, and load or store data. It can also display your stored fies as they are. Hopefully this will be useful in other projects.

Nandetechai

Thank you For Watching

Connect with us.