2nd Stage

Practical Programming Assessment

Data Engineer (f/m/d)

TRUSTED SHOPS

---- START OF PROGRAMMING ASSESSMENT ----

Challenge: Building a Data Pipeline with Luigi (Using JSONPlaceholder API)

You have been tasked with building a data pipeline that fetches data from the JSONPlaceholder API, cleans and transforms it, and stores the processed data in a database. The pipeline should be implemented using the Luigi framework in Python, ensuring proper engineering practices and implementing basic data quality checks.

Requirements:

Create a Luigi Task that fetches data from the JSONPlaceholder API. The task should have the following properties:

- Use the API endpoint https://jsonplaceholder.typicode.com/posts to retrieve data.
- Fetch the data and save it to a local file (e.g., JSON format).
- Implement basic error handling and retries in case of network failures or other errors.
- Ensure that the task is idempotent, meaning it can be run multiple times without duplicating data.

Create a Luigi Task to clean and transform the data. The task should have the following properties:

- Read the data from the file generated by the previous task.
- Implement basic data cleaning and transformation logic specific to the fetched dataset.
- Perform basic data quality checks (e.g., check for missing values, validate data types, etc.).
- Save the cleaned and transformed data to a new file.

(Optional) Create a Luigi Task to load the cleaned data into a database. The task should have the following properties:

- Read the cleaned data from the file generated by the previous task.
- Connect to a database (e.g., SQLite) using a configurable connection string.
- Insert the cleaned data into an appropriate table in the database.

Create a Luigi Workflow that connects the tasks together and represents the data pipeline. The workflow should:

- Specify the dependencies between the tasks to ensure the correct order of execution.
- Handle any failures or errors that may occur during the execution of the pipeline.
- Implement logging and monitoring mechanisms to track the progress and status of the pipeline.

Additional Considerations:

- Implement proper error handling and logging throughout the pipeline.
- Write unit tests for your tasks to ensure they work as expected.
- Focus on implementing the core functionality within the given time frame of two hours. It's okay if you don't complete all the optional steps. The main objective is to demonstrate your understanding of the Luigi framework, data retrieval, and basic pipeline construction.

Please note that the optional steps can be skipped if you find that they exceed the two-hour time limit. Completing the core functionality and demonstrating your understanding of Luigi's key features and proper engineering practices should be your priority. Feel free to adjust the challenge based on your preferences and time constraints.