ETL – Toutour

SDD - 23/10/2025
ALBRECHT Niels
DUC-MARTIN Anaïs
LEFAUCONNIER Lise
POURE Thomas
RAMILISON Hugo

"Moi, j'ai un chien. Mais je n'ai pas le temps de le promener, parce que travail. Comment promener toutou ???"

"To tour with toutour !"

Our startup's mission is to ensure each dog can have at least one walk a day even if its owner is busy. We aim to connect available walkers to dog owners in order for them to take their dog on a walk so that they can keep working.

On our app, dog owners will be able to create their profile as well as each of their dogs' profile. People wanting to spend some time with dogs and at the same time earn some money can become walkers and set up their profile as well. Every time an owner needs his dog to go on a walk, he can create a request on the app. On the other side, every walker can fill in its availabilities and will receive a notification when a request matches its availability. All walks will be saved in the app, to be able to provide a "hot dog" (chien chaud in Quebec) to every owner/dog pair at the end of year. The app will also calculate the amount the walker will earn for each walk. For each walk, the owner will be able to fill in a review about the walker, and the walker a review about the dog.

ETL Pipeline Documentation

1. Extraction

The data source is given by the *Toutour* app users when they register or request a tour. For the demonstration, the data is generated by a python script and stored in a SQL table. Data is received in JSON format (structured as Python dictionaries) from the application. The data will be processed in batches since at launch it is unlikely to be needed to account for numerous users signing up at the same time. Extracted data is presented as CSV files, with one file per SQL table. Data accuracy and completeness are verified before inserting into SQL tables and upon reception from the application, when cleaning the data.

Personal data (e.g., ID, name, age) is involved. Pseudonymization will be needed to remove the link between users and their personal data (address, email...) and still having the possibility to give them personal feedback.

Because our data will only be provided by users, there could be some limitations regarding data accessibility if users forget to fill in the app after their dog went on a walk, or if they don't want to provide some data.

Data updates in the system are triggered by three key events: upon user sign-up (occurring approximately every 50 new users), whenever a tour is requested, and when the "*hot dog resume*" is delivered to the user. To ensure data accuracy and relevance, all affected tables are fully reloaded just before each tour, a process that typically completes in about 15 seconds.

Data will be extracted incrementally at the source. Since it could be costly to reload the whole database each time a new user signs up.

Data volume and velocity depend highly on which data we are manipulating. Registration of new users (owners, walkers, dogs) would have a low to medium volume and are generated daily. Walk requests would need a higher volume with continuous generation and real time handeling. At the end of the day, past walks are registered on our servers; that is a big volume spike. Our system needs to be fully scalable because of the increasing number of active users.

2. Transformation

Here are the data cleaning steps :

- Remove incorrect data types (e.g., strings for age, numbers for dog names).
- Remove rows with missing IDs or primary keys.
- Filter aberrant values (e.g., age, weight).
- Validate email, phone number, and RIB formats.
- Validate consistency between start and end dates.
- Ensure users are adults (≥18 years).

Since Toutour manages personal and financial data, all ETL operations must comply with GDPR principles. Data is collected with user consent, pseudonymized during transformation, and securely stored. Users have the right to access, modify, or delete their data. Data retention is limited to operational needs, and all transfers occur over encrypted connections to ensure confidentiality and integrity.

Because each app user contributes unique data, it's essential to implement a robust system for merging datasets from multiple sources during table uploads. This ensures data consistency, minimizes redundancy, and maintains the integrity of the overall dataset.

Data from the Toutour app is received as JSON objects through the API. To store and process it efficiently in the SQL database, the data is converted into CSV files. Nested JSON fields are flattened, data types are normalized, and all files use standardized encoding and formatting to ensure compatibility during loading.

During the transformation process, all tables are cleaned independently, so there is no dependency between the transformation steps.
Data enrichment is used to create additional calculated fields from existing datasets. Examples include total walking time, average distance per dog, or total earnings per walker. These new metrics enhance the analytical value of the data and support features such as the "Hot Dog Resume." Enrichment occurs during transformation without modifying original user data.

Audit logs recording every update on the data frames. Furthermore, each update would not destroy the previous data but simply create a new version, and the old versions would be stored with timestamps to easily monitor the updates timeline. For sensible data such as paiments on the app, we need to be sure that the logs cannot be modified and are exact.

3. Loading

Data is meant to be loaded on mobile devices. Data is loaded hourly. Data is appended to existing datasets, as it is sourced from users. The schema must be compatible with iOS and Android platforms. Data consistency and integrity are ensured through pre-loading data cleaning processes. Data is loaded into a mobile-friendly SQLite database or API endpoint accessible by the Toutour app.

Data loading must be fast enough to ensure a smooth and responsive user experience, particularly during walk selection and request confirmation. The system should support near-instantaneous data access (under 2–3 seconds) even under peak load.

To achieve this, data will be pre-cleaned and indexed before loading, and caching mechanisms will be implemented to reduce redundant queries. Incremental loading and asynchronous updates will also help maintain performance as the user base grows.

Schema evolution is managed through versioned migration scripts, ensuring backward compatibility on mobile devices. For instance, if we were to account for new animals, a feature should be added to categorize the different animal types.

Historical data and versioning are maintained for auditing purposes. Each update generates a new version of the affected records, while older versions are preserved with timestamps to allow full traceability of changes. This ensures data integrity and supports GDPR compliance.

Monitoring will include automated alerts on ETL failures and daily success reports via logging dashboards (e.g., Grafana or Airflow).

4. ETL pipeline preview

Full pipeline here : https://github.com/anaiisdcm/ETL-Toutour