Santropol Roulant's Meals on Wheels Dashboard

A Technical Wiki

Last Update: July 20, 2024

Authors: Jared Balakrishnan, Moiz Shaikh

Introduction

This document serves to explain the technical specifications of the solution built by the authors as part of their community project as graduate students at McGill University carried out at Santropol Roulant, an Montréal-based non-profit organization focusing on sustainable food systems.

The authors of this document worked on building an analytics stack that would serve a user-friendly, detailed dashboard chronicling detailed key performance indicators centered around the clients and volunteers working with Santropol Roulant's Meals on Wheels Program.

The objective of this project was to streamline and automate the calculation and circulation of key performance indicators thereby enabling the employees in charge of the Meals on Wheels program to bolster fundraising efforts, continue to foster relationships with donors and improve the allocation of volunteering resources to the program.

Technology Stack

System Requirements

In order to be able to run this project successfully, the minimum system requirements include:

- No specific Operating System; any computer should be able to run this project.
- Web Browser: While Google Chrome is recommended, this project should be able to run on any web browser.
- MySQL: MySQL (Version 8 and above) is required in order to host all of the data related to this project.
- Python: Python (Version 3.6 and above) along with the listed libraries is required to run this project.
- Node: NodeJS (Version 20) is required in order for Evidence to be able to run successfully and make changes to the underlying source code. However, this is NOT a requirement for anyone just wanting to use the dashboard.

For developers seeking to maintain, modify or extend the source code - the requirements.txt file in the GitHub repository has all of the python packages listed, in addition to a detailed listing of all the source code files used in the project.

Summary of Tools Used

The tools used to build this project are summarized in the following table:

Area of Work	Tool Used	Libraries Used	Open-Source?
Frontend Development	Evidence Markdown		Yes
Backend Development	Python	pandas, pyairtable, sqlalchemy, pymysql	Yes

Database Management	MySQL + Planetscale		Yes

Each one of the components in the table is explained in more detail in the subsequent subsections.

Frontend Development

The frontend development for this project was carried out using a brand new, fully open sourced form of markdown called Evidence Markdown that enables the construction of complete data projects using just SQL and Markdown.

Backend Development

The backend development for this project was carried out entirely in Python. Python was chosen for this project considering the availability of open-source libraries that make it possible to easily work with large amounts of data.
This project also involves the use of a few Python libraries:

- pyairtable: pyairtable is an open-source community designed Airtable API client that serves to enable Python developers to be able to access data that is stored across Airtable databases considering the fact that the official Airtable API only has a node JS powered client.
- pandas: Pandas is used to convert all of the API response JSONs into neat, tabular pandas data frames prior to cleaning and further processing, the output of which is then normalized into tables in a MySQL database.
- sqlalchemy + pymysql: The combination of the SQLAlchemy and pymysql libraries are used to set up connections to the MySQL database directly from the Python script and allow for query execution and connections.

Database Management

- MySQL: The KPIs and other metrics, as well as the DDL implementation of the tables used to hold the output of the backend development processes

are implemented in MySQL. For the purposes of this project, these MySQL databases were hosted locally, as well as on a server hosted by Planetscale. **In the long run, the client will be hosting this data onto their own cloud infrastructure, which is currently being worked on in their internal tech roadmap.**

- Planetscale: Planetscale was used to host the MySQL database so that both of the team members could connect and query from a singular source of truth. **It is to be noted that the usage of Planetscale will be ceased upon the handover of the project artifacts of the client owing to the fact that they have their own hosting infrastructure (both on-premises and cloud).**

Data Sources Used

Airtable

Individuals interested in volunteering at Santropol Roulant are required to fill out a form available on their website. This form is an Airtable form, and the responses to this form are stored in Bases (Airtable parlance for database tables) that reside in a central Airtable account owned by the organization.

All of the volunteer registration data therefore resides, and is obtained from Airtable for this project.

Sous-Chef

Sous-Chef is an open-source Python application for Meals on Wheels programs service management licensed under “Affero General Public License 3”. It was developed by Savoir Faire Linux, one of the largest open-source companies operating out of Canada in partnership with Maison du logiciel libre as well as the local Montreal Python developer community.

The backend of Sous-Chef, implemented in MariaDB, contains all of the data relating to the vulnerable and elderly clients receiving meals through Santropol Roulant's Meals on Wheels program.

Architecture

The solution architecture for this project is summarized in the below shown diagram, and explained in detail in the following sections:

Data Extraction

As mentioned above, there are two main sources of data used in the project:

- Airtable, which hosts the volunteer registration-side data.
- Sous-Chef, which hosts the client-side data.

Airtable

The data from Airtable is extracted using pyairtable, the Python-based Airtable API client through the following steps:

- A personal access token is first generated in the administrator-owned Airtable instance. This token is visible only to individuals with administrator access rights to Santropol Roulant's Airtable account. It is obfuscated from the source code through the use of environment variables local to the machine the code is executed from. This personal access token also only offers read access to selected data tables in the Airtable account, meaning access to data is contained in the event a bad actor gains access to the token.

- The Airtable-generated API field guide is then used as an anchor to sanitize the response from the API to only include those fields that are relevant to Santropol Roulant's Meals on Wheels Program.

Sous Chef

On the other hand, Sous-Chef data is already collected, organized and stored in a MySQL database that is hosted through an on-premises server located in the basement of the Santropol Roulant building. For the purposes of this project, a copy of the database was ported over to a Planetscale instance to enable both developers to have access to the same tables. Additionally, both developers also had access to a copy of the database hosted on their local machines.

Data Processing

- The Airtable API response from the above step is then converted into a pandas dataframe, after which a series of steps are applied to clean it before the dataframe can be passed to a MySQL database from where it can be queried and loaded to the frontend (Evidence in this case).
    - The API response is normalized into 9 different tables, because of which the cleaning process differs from table to table. The source code has comments detailing the cleaning processes carried out for each table.
- The Volunteer Registration database is observed to have over 50 fields, which results in the formation of a very large table. Considering the large number of fields in addition to the redundancy in terms of information available, a star schema design comprising 9 tables (1 fact table + 8 dimension tables) was adopted to better organize and store the data:
    - FACT_VOLUNTEER_CENTRAL: This is the main fact table that contains important, overarching information about registered volunteers.
    - DIM_VOLUNTEER_FORM: This dimension table contains a copy of the form filled out by the volunteers as-is.

    - DIM_VOLUNTEER_CONTACT: This dimension table contains contact information relating to each registered volunteer.
    - DIM_VOLUNTEER_ICE: This dimension table contains information about the individuals to be contacted in case of an emergency to a registered volunteer.
    - DIM_VOLUNTEER_SKILLS: This dimension table contains the skill information sought out by Santropol Roulant that the volunteers marked themselves as having.
    - DIM_VOLUNTEER_PROGRAM_PREFERENCES: This dimension table contains information about each registered volunteer's interest in each of the volunteering programs offered by Santropol Roulant.
    - DIM_VOLUNTEER_DELIVERY_SHIFT_AVAILABILITY, DIM_VOLUNTEER_KITCHEN_SHIFT_AVAILABILITY and DIM_VOLUNTEER_DELIVERY_PREFERENCES: These three dimension tables contain information restricted to delivery and kitchen shifts associated with the Meals on Wheels' Delivery program.
    - ***Data Dictionaries (description of each field along with data types as well as any other important information) for each one of these 9 tables is made available in the Excel spreadsheet titled "Volunteer-Data-Normalized-Data-Dictionary.xlsx" in the "Data Dictionaries" folder found within this submission.***
- The Sous Chef data required no subsequent processing considering the fact that the existing application code already takes care of any and all cleaning tasks necessary.

Data Loading

- After the processing steps described above, the Airtable data is transferred to the same MySQL database containing the Sous Chef data.

Dashboard Development and Rendering

- Evidence allows MySQL databases as a valid data source, and therefore the MySQL database containing the processed data is then connected to Evidence.
- All of the subsequent KPI and data visualization development is carried out through a combination of DuckDB SQL dialect and Markdown.

A comprehensive user manual, detailed listings of the KPIs as well as descriptions of each can be found directly on the dashboard in both English and French.

A Note on Hosting

During the development of this project, all hosting was carried out on the developers' local machines since the client did not want their data being surfaced anywhere.

This project is designed to be easily hosted; for an end-user to be able to use the dashboard and be able to use the insights provided, all that needs to be done is for the Evidence project to be hosted on a server. An end-user can simply navigate to the webpage, nothing else is necessary.

Currently, the Roulant has a server in its basement but is transitioning to Google Cloud. The plan is to initially host this dashboard on their local server, before moving it to the Google Cloud instance. The final steps to be taken, however, will depend on the results of the final hand-off conversation between the developers and the client.

Appendix A: Entity-Relationship Diagrams of the Database

Sous-Chef Data