

Simple Flight Data Service

Your assignment is to create a simple service which can perform different tasks, computations and queries on top of Federal Aviation Administration (FAA) data. The essential FAA data are included with this assignment and are in Parquet format.

The service should have either REST or a gRPC API - pick the one which you are most comfortable with. Note: for sake of brevity, this assignment refers to both REST API resources and gRPC methods as endpoints.

Together with the service, you need to provide either end to end tests for the endpoints or at least a standalone script that exercises the different endpoints and their functionality. The script can be even shell script that uses curl or [grpcurl](#) to make the necessary calls.

Create a new GitHub repository for this assignment; the repository should be self-contained and include the FAA data, code of the service and code to test the service.

Constraints, clarifications and hints

- Your solution **can use any open source packages**.
- Your solution **can keep all the data in memory**.
- Your solution **must not depend on any external processes** such as database servers (PostgreSQL, MySQL). It must be fully contained within the Python process.
- You do not need to worry too much about the format in which the results are returned by the service - make the endpoints return results in CSV if you do not want to spend too much time figuring out the right JSON or protocol buffer message.

Tasks

1 Create a new Python 3 project for your assignment

Do this according to your best judgment, knowledge and practices.

2 Create the service that owns the FAA data

The service runs in its own server and must have either REST API or a gRPC API. When the service starts up, it should read and load all the FAA data.

For starters, your service should have a single endpoint through which clients can get information about the loaded data sets. For each of the FAA data file, your service should return the following:

- Name of the data file
- List of columns
- Total number of rows

3 List known aircraft models

Enhance your service with a new endpoint, which the client can call with no parameters and that will return all known aircraft models, their manufacturer and number of seats.

All data needed for this endpoint is in the `aircraft_models` data file.

4 List active aircrafts by manufacturer and model

Enhance your service with a new endpoint, which the client can call with two parameters:

- Aircraft manufacturer
- Aircraft model

Given this input, your service must return all active aircrafts of the selected model and manufacturer. For each found aircraft, the endpoint must return:

- Manufacturer
- Model,
- Number of seats
- Serial number
- Registrant name
- Registrant county

All data needed for this endpoint is in the `aircraft` and `aircraft_models` data files.

Hints:

- The `aircraft` data file has a `status_code` column. The value "A" denotes that the aircraft is active
- The `aircraft` data file has `name` and `county` columns. These contain the registrant's name and county.

Advanced Tasks

1 Generate report of active aircraft models by county

Enhance your service with a new endpoint, which the client can call with no parameters and that will return summary counts of active aircraft models and their manufacturer by the county in which those aircrafts are registered.

All data needed for this endpoint is in the `aircraft` and `aircraft_models` data files. See hints from previous tasks to get a better idea what columns you will need to work with.

2 Generate pivot report of active aircraft models by count

Enhance your service with a new endpoint that returns the same data as in the previous task, however with differently formatted results.

The result should have one row for each aircraft model and its manufacturer and the aircraft count per county should be in a column dedicated to that county. If the aircraft model is not registered in the county, then the value is NULL.

An example output:

Manufacturer	Model	CA	TX	WA
Boeing	747-406	10	20	15
Kawasaki	KV107-11A	1	NULL	2

3 Allow clients to run SQL on top of the FAA data

Add a new endpoint through which clients can submit SQL to run on top of the FAA data. We are looking for rich SQL support with functions, casts, joins and aggregations.

Optional Tasks

- 1 Make all your code typed and use `mypy` for type checking
- 2 Create Docker image for your service