Tech Assignment - 2 - API server generation and setup

Wednesday, August 20, 2025

6:26 PM

**==== About the BrainBox API server ========**

**BrainBox API server requirements**

The API provides two main endpoints:

1. **File Upload Endpoint**: Accepts CSV file uploads and stores them in an S3 bucket, then triggers a Glue crawler to make data queryable via Athena (database: bills\_db, schema: custom\_csv)
2. **Validation Endpoint**: Validates CSV structure and returns a MissingField exception for invalid data

**Required CSV Schema**

Valid CSV files must contain exactly these columns with specified data types:

* bill\_id (string)
* meter\_id (integer)
* usage\_type (string)
* building\_id (integer)
* start\_date (ISO date format)
* end\_date (ISO date format)

**BrainBox API server iterative version history (4)**

* I generated / developed with AI LLMs just for this tech assignment testing purpose as Anirban suggested, and keep improving and fixing issues during this tech assignment testing.
* **V1: Basic functionality POC (not working) -** using fastAPI and uvicorn for this API app server
* **V2: Internal iteration & curl test ok** — interacted, updated API endpoints, tested by curl commands, works
* **V3: Added error handling for schema and basic AWS service local simulation test support** - such as write artifacts to uploads/ and glue marker to glue/)
* **V4 - Added supports for AWS 'Athena" simulator via SQLite**
  + Added simulation and mirror via local DB (SQLite) that populated uploaded CSV into DB for testing, database name: bills\_db, table name: custom\_csv;
  + Added new 'query' API endpoint to support SQL query call (optional)
    - New endpoint workflow: '/validate' -> '/upload' -> stores the CSV -> **triggers Glue marker -> loads rows into DB -> SQL query, example**:
      * SELECT \* FROM custom\_csv;

**Major components under the API server**

* Uvicorn - Web app server
* FastAPI - REST API implementation
* SQLite - Database

**(Ref) Artifacts under the API server after CSV file uploaded: /local\_state/glue:**

* **Location:** local\_state/glue/
* **Artifacts:**
  + local\_state/glue: “fake Glue crawler” markers—simple files we drop every time /upload triggers the crawler so you can verify the side-effect without AWS.
  + Local\_state/uploads: stores the uploaded CSV file
  + Local\_state/db: stores the database file with populated data from the uploaded CSV (simulate Athenna by sqlite)
* **File name pattern for Glue crawler**
  + Format / Pattern: bills\_crawler\_\_<unix\_timestamp>\_\_<uuid>.marker
    - Example: bills\_crawler\_\_1724025605\_\_9f1a3f1c6f0e4a8dbe5a9c0c8b3e6d8b.marker
    - bills\_crawler → the crawler name
    - 1724025605 → when it was triggered (Unix seconds)
    - <uuid> → unique run id

**====================**

* **Generated a local API functionality server for this tech assignment testing via LLMs (ChatGPT)**

Linux‑friendly package you can download and run locally (no AWS needed).

**Download:** brainbox\_qa\_local\_api\_v2.zip

What you’re getting:

* FastAPI app with two endpoints: /upload (stores CSV locally + simulates Glue trigger) and /validate (checks schema, returns MissingField on error).
* Pure‑Python **unit tests** (validators + local “AWS” facade).
* **Integration tests** using FastAPI TestClient (no server process required).
* **Sample CSVs** for happy/negative paths.
* A **strategy doc** with AWS pseudo‑code you can include for deliverable #3.
* README.md with exact Ubuntu steps, plus Makefile shortcuts.

* **Ubuntu - Setup for the API server**
  + **Prepare env** 
    - Setup Python support

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ sudo apt update

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ sudo apt install python3

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ sudo apt install python3-pip

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ sudo apt install python3.12-venv

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ python3 --version

**Python 3.12.3**

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ pip3 --version

**pip 24.0** from /usr/lib/python3/dist-packages/pip (python 3.12)

* **Install necessary packages** 
  + sudo apt install unzip
  + Sudo apt install zip
  + sudo apt install jq

* **Create new folder for the local API server** 
  + cd /opt
  + sudo mkdir brainbox\_qa\_local\_api
  + sudo chmod 777 brainbox\_qa\_local\_api

* **Copy and unzip the new API server:**
  + unzip brainbox\_qa\_local\_api\_v4.zip
  + Copy all files under folder: /opt/brainbox\_qa\_local\_api

* **Install Python packages / modules for API server:**

cd /opt/brainbox\_qa\_local\_api

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ python3 -m venv .venv && source .venv/bin/activate

(.venv) ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ pip3 install -r requirements.txt

* Note:
  + Updated the requirements.txt to add 'python-multipart' as another required package due to error:"RuntimeError: Form data requires "python-multipart" to be installed.".

* **Start API Application Server:**

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ uvicorn app.main:app --reload --port 8000

INFO: Will watch for changes in these directories: ['/opt/brainbox\_qa\_local\_api']

INFO: Uvicorn running on <http://127.0.0.1:8000> (Press CTRL+C to quit)

INFO: Started reloader process [13507] using StatReload

INFO: Started server process [13509]

INFO: Waiting for application startup.

INFO: Application startup complete.

* Check App server started:

ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api$ ps -ef | grep Uvicorn

ghu 13512 645 0 16:50 pts/0 00:00:00 grep --color=auto Uvicorn

* **Test with curl commands to verify App server functionality works:**

(.venv) ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api\_v4$ **curl -i -X POST -F "file=@sample\_data/valid\_bills.csv"** [**http://loc**](http://loc)

**alhost:8000/validate**

HTTP/1.1 200 OK

date: Tue, 19 Aug 2025 16:21:36 GMT

server: uvicorn

content-length: 133

content-type: application/json

{"message":"CSV file is valid.","columns":["bill\_id","meter\_id","usage\_type","building\_id","start\_date","end\_date"],"rows\_checked":1}

(.venv) ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api\_v4$ **curl -i -X POST -F "file=@sample\_data/valid\_bills.csv"** [**http://localhost:8000/upload**](http://localhost:8000/upload)

HTTP/1.1 200 OK

date: Tue, 19 Aug 2025 16:22:43 GMT

server: uvicorn

content-length: 183

content-type: application/json

{"status":"ok","stored":"local\_state/uploads/valid\_bills.csv","crawler\_marker":"local\_state/glue/bills\_crawler\_\_1755620564\_\_fb1fc83dbd374efcbc757ef2fd376cbe.marker","rows\_inserted":1}(.venv)

(.venv) ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api\_v4$ **curl -s -X POST** [**http://localhost:8000/athena/query/**](http://localhost:8000/athena/query/) **-H "Content-Type: application/json" -d '{"sql":"SELECT \* FROM custom\_csv LIMIT 10"}' | jq**

{

"columns": [

"bill\_id",

"meter\_id",

"usage\_type",

"building\_id",

"start\_date",

"end\_date"

],

"rows": [

[

"b1",

100,

"kwh",

1,

"2024-01-01",

"2024-01-31"

]

],

"rowcount": 1

}

(.venv) ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api\_v4$ **curl -i -X POST -F "file=@sample\_data/invalid\_missing\_field.csv"**

[**http://localhost:8000/upload**](http://localhost:8000/upload)

HTTP/1.1 422 Unprocessable Entity

date: Tue, 19 Aug 2025 16:33:48 GMT

server: uvicorn

content-length: 86

content-type: application/json

{"error":"InvalidSchema","missing":["building\_id","start\_date","end\_date"],"extra":[]}

(.venv) ghu@GHThinkPad1:/opt/brainbox\_qa\_local\_api\_v4$ **curl -i -X POST -F "file=@sample\_data/invalid\_missing\_field.csv"** [**http://localhost:8000/validate**](http://localhost:8000/validate)

HTTP/1.1 422 Unprocessable Entity

date: Tue, 19 Aug 2025 16:34:30 GMT

server: uvicorn

content-length: 86

content-type: application/json

{"error":"InvalidSchema","missing":["building\_id","start\_date","end\_date"],"extra":[]}