

Assignment

Problem Statement: Design a restful API that will perform the following operations:

- ◆ Take CSV file as input (*find it in the attachment*)
- ◆ Process the data.
- ◆ Save the processed data into a database. ◆ Return the output in json format.

Dataset Details: Dataset contains the account details of customers and it consists of three columns:

- ◆ **txn date** : date of transaction.
- ◆ **narration** : a string that contains transaction data.
 - It is in the below format:
 - txn/acc_number/rrn/ifsc_number/bank_name/acc_holder_name/txn_type
 - E.g: TXN/XXXX9695/RRN:2203069/78281172/Punjab National Bank/SHARAN KUMAR/IMPS

TXN	Constant Keyword
XXXX9695	Constant Number
RRN: 2203069	RRN Number (Use this as primary key) <i>use only number (2203069)</i>
78281172	Account Number
Punjab National Bank	Bank Name
SHARAN KUMAR	Account Holder Name
IMPS	Transaction Type

- ◆ **Amount** : indicates transaction amount

API Details:

Method	Endpoint	Description
POST	/post_data	This will take the data, and saves it in the database after processing and return a successful response.
GET	/records	This api call should return number of rows present in your db in json format.

GET	/banks	This api call should return number of unique banks present in your db in json format.
GET	/from_date/to_date	This api call will take 2 query parameters as input i.e, from date and to date and will return number of transactions occurred during that interval.
GET	/customer_names	This api should return names of all customers in Camel Case format. (e.g, Ram Mishra)
GET	/transactions_summary	This api call should return number of transactions based on its type. E.g, { 'IMPS' : 10, 'NEFT' : 15 }
GET	/transaction_amount_summary	This api call should return total amount of transactions based on its type. E.g, { 'IMPS' : 12,265, 'NEFT' : 10,560 }
GET	/total_transaction_amount	This api call should return total transaction amount.

Point To Be Noted:

- Use rrn as primary key in the database.
- There can be a duplicate entry as well, so take care of that.
- Store Customer Name in Camel Case Format.
- Store amount as float number.
- Store transaction in proper date format.
- Create a suitable data model to persist the data.
- Firstly, to seed the data we will post the data and then we will hit subsequent get requests.
- Think database modeling in a way to accommodate ingesting 10M records everyday

Technology Stack Allowed:

- Preferably - flask/fastapi
- Database (Any of the choices)

Submission:

- Specify requirements.txt file for installing dependencies.
- Specify commands to start the server.
- Specify all the API endpoints.
- Try to follow OOPS concepts while writing the code.
- Upload your code in a github repository and share the link with us